TUGAS AKHIR - TM091486

PERANCANGAN *INTEGRATED REAL-TIME MONITORING SYSTEM* UNTUK MOBIL LISTRIK EZZY ITS

GRANGSANG SOTYARAMADHANI
2109100044

Dosen Pembimbing
Dr. Muhammad Nur Yuniarto, ST

PROGRAM STUDI SARJANA
JURUSAN TEKNIK MESIN
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2014

TUGAS AKHIR - TM091486

DESIGN OF INTEGRATED REAL-TIME MONITORING SYSTEM FOR ELECTRIC VEHICLE EZZY ITS

GRANGSANG SOTYARAMADHANI
2109100044

Faculty Advisor
Dr. Muhammad Nur Yuniarto, ST

DEPARTMENT OF MECHANICAL ENGINEERING
FACULTY OF INDUSTRIAL TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA 2014

LEMBAR PENGESAHAN
TUGAS AKHIR (TM091476)

PERANCANGAN *INTEGRATED REAL-TIME MONITORING SYSTEM* UNTUK MOBIL LISTRIK EZZY ITS

TUGAS AKHIR
Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik pada
Program Studi S-1 Jurusan Teknik Mesin
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Disusun Oleh :
GRANGSANG SOTYARAMADHANI
NRP. 2109100044

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr. Ir. M. Nur Yuniarto
   NIP. 197506301998621001 .....(Pembimbing)
2. Dr. Eng. Sutikno, ST, MT
   NIP. 197004121997032003 .....(Penguji I)
3. Indra Sidharta, ST, MSc
   NIP. 198006192006041004 .....(Penguji II)
4. Wahyu Wijanarko, ST, MSc. Eng
   NIP. 198202092012121001 .....(Penguji III)

SURABAYA
DESEMBER 2014

PERANCANGAN *INTEGRATED REAL-TIME MONITORING SYSTEM* UNTUK MOBIL LISTRIK EZZY ITS

Nama Mahasiswa : Grangsang Sotyaramadhani
NRP : 2109 100 044
Jurusan : Teknik MesinFTI-ITS
Dosen Pembimbing : Dr. Muhammad Nur Yuniarto

Abstrak

Mobil listrik Ezzy ITS yang telah dibuat dan sedang dikembangkan oleh Institut Teknologi Sepuluh Nopember (ITS) Surabaya memiliki tiga sistem utama. Tiga sistem utama tersebut adalah sistem penggerak, sistem penyimpanan energi, dan sistem aksesoris. Ketiga sistem tersebut saat ini belum terintegrasi menjadi satu. Saat ini prosedur pengaturan dan *monitoring* untuk masing-masing sistem tersebut harus dilakukan secara terpisah dengan *tools* yang berbeda-beda. Sehingga diperlukan sebuah sistem baru yang mampu mengintegrasikan ketiga sistem tersebut. Oleh sebab itu pada penelitian tugas akhir ini akan dilakukan perancangan *Integrated Real-Time Monitoring System* yang bernama IVC (*In Vehicle Computer*).

Perancangan IVC akan disesuaikan dengan kebutuhan dan spesifikasi dari ketiga sistem utama pada Ezzy ITS. Raspberry Pi Model B dan STM 32 F3 Discovery Board merupakan *hardware* yang berfungsi sebagai *central processing unit* pada IVC. Pada tugas akhir ini juga akan dirancang *software* khusus. *Software* tersebut akan ditanamkan pada *hardware* sehingga data yang didapat dari ketiga sistem pada Ezzy ITS dapat diolah. Hasil olahan data kemudian akan ditampilan dalam sebuah *graphical user interface* (GUI) pada sebuah LCD agar dapat dimengerti oleh pengemudi. Selain itu pada IVC untuk Ezzy ITS juga digunakan protokol komunikasi data CAN (*Controller Area Network*) sehingga jumlah data yang dapat dimonitor cukup banyak.

Pada tugas akhir perancangan Integrated Real-Time Monitoring System atau disebut juga IVC ini didapatkan hasil Arsitektur IVC yang sesuai dengan sistem Mobil Listrik Ezzy ITS. Selain itu juga telah dapat ditentukan Hardware untuk IVC yang mampu untuk mengelola sinyal-sinyal dari sistem Mobil Listrik Ezzy ITS untuk ditampilkan kepada pengemudi secara real-time. Sehingga secara garis besar dapat dikatakan bahwa Sistem IVC yang dirancang pada tugas akhir ini telah berhasil.

Kata Kunci : Integrated Real-Time Monitoring System, mobil listrik, CAN, STM 32 F3 Discovery

DESIGN OF INTEGRATED REAL-TIME MONITORING SYSTEM FOR ELECTRIC VEHICLE EZZY ITS

Name : **Grangsang Sotyaramadhani**
NRP : **2109 100 044**
Department : **Mechanical Engineering**
Faculty Advisor : **Dr. Muhammad Nur Yuniarto**

Abstract

Electric vehicle Ezzy ITS which has built and been developing by Institute Technology of Sepuluh Nopember (ITS) Surabaya, has three main systems. They are drivetrain system, energy storage system and accessories system. Currently, third of main systems have not been integrated each others. Setting up and monitoring procedures should be done separately by using different tools. Furthermore, it is needed a new system which is capable to integrating the third of systems. On this final project research will be designed of Integrated Real-Time Monitoring System, known as IVC (In Vehicle Computer).

Design of IVC will be represented by requirement and specification of third of the main systems for Ezzy ITS. Raspberry Pi Model B and STM 32 F3 Discovery Board are hardwares which having function as central processing unit on IVC. This final project research will be designed of specific software. It will be installed on hardware, so that data which getting from third of the main systems can be processes on Ezzy ITS. The outputs will be displayed on a Graphichal User interface (GUI) of a LCD, so that the data can be understood by the drivers. Moreover, Ezzy ITS's IVC was used CAN (Controller Area Network) communication protocol, so that a lots of data can be monitored.

On final project research, design of Integrated Real-Time Monitoring System, or known as IVC, can be produced appropriate IVC architecture on electric vehicle Ezzy ITS. Moreover, hardware has been determined for IVC which is capable to processing signals from electric vehicle Ezzy ITS system. It will be displayed real

time for drivers. With the result that design of IVC system has been succeed on this final project research.

Keywords : Integrated Real-Time Monitoring System, Electric vehicle, CAN, STM 32 F3 Discovery

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa karena atas anugerah, berkah dan hidayah-Nya laporan Tugas Akhir yang berjudul "Perancangan *Integrated Real-Time Monitoring System* untuk Mobil Listrik Ezzy ITS" ini dapat diselesaikan.

Dalam kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sedalam-dalamnya kepada:

1. Bapak Dr. Muhammad Nur Yuniarto selaku dosen pembimbing yang telah banyak memberikan arahan, bimbingan serta pelajaran selama pembuatan dan penyelesaian mobil juga selama perlombaan berlangsung.
2. Orangtua serta seluruh keluarga yang telah banyak memberikan dukungan moral.
3. Seluruh tim *ITS Solar Car Racing Team* dan tim Molina ITS yang telah banyak membantu dalam pembuatan Tugas Akhir ini.
4. Siti Choirun Nisa yang telah membantu dari awal sampai akhir pembuatan proposal Tugas Akhir ini.

Penulis menyadari bahwa laporan tugas akhir ini masih belum sempurna, baik dari analisis yang penulis lakukan maupun dalam penulisan laporan. Semoga laporan ini dapat memberikan manfaat bagi pembaca pada umumnya dan penulis pada khususnya.

Surabaya, 22 Desember 2014

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1. Latar Belakang

Institut Teknologi Sepuluh Nopember Surabaya (ITS) sebagai salah satu perguruan tinggi negeri berbasis teknologi terbaik di Indonesia memiliki peranan cukup penting dalam perkembangan teknologi di Indonesia. Salah satu bidang teknologi yang dikuasai oleh ITS adalah adalah teknologi otomotif. Pada Tanggal 26 Januari 2013, ITS telah meluncurkan mobil listrik yang diresmikan oleh Menteri Pendidikan dan Kebudayaan Indonesia. Mobil listrik berjenis *city car* milik ITS tersebut dinamai Ezzy (EC) ITS. Hingga saat proposal tugas akhir ini ditulis, ITS telah meluncurkan 2 mobil listrik yaitu Ezzy ITS ver 1.0 dan Ezzy ITS ver 1.1.

Di dalam Ezzy ITS terdapat tiga sistem utama yaitu sistem penggerak, sistem penyimpanan energi, dan sistem aksesoris (*auxiliary system)*. Sistem penggerak terdiri dari *electric motor* dan *controller*. Kemudian sistem penyimpanan energi terdiri dari *battery* dan *battery management system* (BMS). Selanjutnya sistem aksesoris terdiri dari sistem kelistrikan untuk lampu dan sistem indikator keamanan pengemudi. Saat ketiga sistem pada Ezzy ITS tersebut belum terintegrasi menjadi satu. Sehingga prosedur pengaturan dan *monitoring* untuk masing-masing sistem tersebut harus dilakukan secara terpisah dengan menggunakan *tools* yang berbeda–beda.

Dalam usaha mengembangkan dan menyempurnakan Ezzy ITS diperlukan sebuah sistem baru yang mampu mengintegrasikan ketiga sistem tersebut. Sistem baru yang dimaksud adalah *integrated real-time monitoring system* (IRTMS). IRTMS dapat disebut sebagai *main monitoring system* yang bertugas mengintegrasikan sistem *monitoring* beberapa *sub system*, dalam kasus ini adalah sistem penggerak, sistem penyimpanan energi, dan sistem aksesoris. IRTMS untuk mobil listrik Ezzy ITS selanjutnya

disebut *In Vehicle Computer* (IVC). IVC terdiri dari tiga komponen utama yaitu *Hardware*, *Software*, dan *Graphical User Interface* (GUI). *Hardware* dan *software* dari IVC berfungsi untuk menerima, mengirim, dan memproses data dari masing-masing sistem melalui protocol komunikasi data. GUI berupa tampilan informasi yang dapat dibaca oleh pengendara berdasarkan data yang telah diproses oleh *hardware* dan *software*. Fokus dari tugas akhir ini adalah sebagai berikut, merancang arsitektur IVC untuk Ezzy ITS, membuat IVC untuk Ezzy ITS, membuat *software* sekaligus GUI IVC untuk Ezzy ITS.

1.2. Rumusan Masalah

Permasalahan dalam tugas akhir ini adalah:

1. Bagaimana rancangan arsitektur IVC yang sesuai untuk Ezzy ITS
2. *Hardware* apa yang sesuai dengan arsitektur IVC untuk Ezzy ITS
3. Bagaimana rancangan *software* IVC untuk Ezzy ITS
4. Bagaimana desain GUI IVC untuk Ezzy ITS

1.3. Tujuan Tugas Akhir

Tujuan tugas akhir ini adalah:

1. Merancang arsitektur IVC yang sesuai untuk Ezzy ITS
2. Menentukan *hardware* yang sesuai untuk IVC pada Ezzy ITS berdasarkan arsitektur IVC yang telah dirancang
3. Merancang *software* IVC untuk Ezzy ITS
4. Mendesain GUI IVC untuk Ezzy ITS

1.4. Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah:

1) Arsitektur IVC untuk Ezzy ITS yang akan dirancang dibatasi hanya untuk mengintegrasikan sistem *monitoring* dari tiga sistem utama dengan data atau parameter sebagai berikut:

   a. Sistem Penggerak (*Electric Motor* dan *Controller*) : *RPM* dan Kecepatan

   b. Sistem Penyimpanan Energi (*Battery* dan *Battery Management System* BMS) : SoC, *Pack Voltage, Lowest Cell, 12V Supply, Pack Current, Average Temperature, Average Cell Voltage.*

   c. Sistem Pendukung : Lampu Kota, Lampu Jauh, Lampu Dekat, Lampu Sein, Indikator Hazard, dan Indikator Rem Tangan.

2) GUI IVC untuk Ezzy ITS dibatasi berupa tampilan status dari masing-masing sistem berdasarkan data yang disediakan oleh ketiga sistem utama.

1.5. Manfaat Tugas Akhir

Manfaat dari tugas akhir ini adalah:

1) *Monitoring system* pada mobil listrik Ezzy ITS dapat terintegrasi menjadi satu.
2) Pengemudi mobil listrik Ezzy ITS dapat memperoleh informasi performa Ezzy ITS secara akurat dan *real-time*.

1.6. Sistematika Laporan

Sistematika penulisan dibagi dalam beberapa bab sebagai berikut:

1) Bab 1 Pendahuluan, bab ini berisi latar belakang dari penelitian, rumusan masalah, batasan masalah, manfaat, dan sistematika penulisan laporan
2) Bab 2 Tinjauan Pustaka dan Dasar Teori, bab ini berisi dasar-dasar ilmu yang mendukung pengerjaan tugas akhir
3) Bab 3 Metodologi Penelitian, bab ini berisi urutan langkah-langkah proses perancangan sistem IVC
4) Bab 4 Analisa dan Pembahasan, bab ini berisi detail proses perancangan dan pengujian sistem IVC
5) Bab 5 Kesimpulan dan Saran, bab ini berisi kesimpulan dari penelitian yang telah dilakukan serta saran-saran untuk pengembangan penelitian ini di masa mendatang

BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Pendahuluan

Derek S. Wright (1990) dalam paper-nya menuliskan bahwa aplikasi *on-broad vehicle system* tersekmentasi menjadi tiga ruang lingkup yaitu:

1) **Kontrol dan performa teknis dari kendaraan**
   Pada lingkup ini *on-board vehicle system* difungsikan untuk mengontrol beberapa sistem dasar pada kendaraan seperti *Power Train System*, *Braking System*, *Steering System*, *Suspensions*, dan *Vehicle Interiors* untuk menjaga performa teknis dari kendaraan.

2) **Interaksi dari kendaraan dengan lingkungan**
   On-board vehicle system juga dapat difungsikan untuk mengintegrasikan kendaraan dengan lingkungan sekitar. Sistem navigasi dan penunjuk posisi kendaraan adalah bentuk implementasi untuk ruang lingkup ini.

3) **Efisiensi dan keefektifan operasional dari kendaraan / sistem**
   Seiring berkembangnya sistem komunikasi *mobile* maka operasional dalam berkendara menjadi lebih efisien dan efektif dengan menggunakan *on-board vehicle system* yang terintegrasi dengan sistem komunikasi.

Derek menyimpulkan bahwa *on-board system* telah sangat berkembang sehingga mampu meningkatkan performa kendaraan dari berbagai aspek.

2.2. Gambaran Umum IRTMS (*Integrated Real-Time Monitoring System*)

Pada umumnya IRTMS memiliki komponen – komponen utama sebagai berikut:

1) CPU (*Central Processing Unit*)
   CPU berfungsi untuk memproses segala informasi yang diterima melalui *Communication Module* serta membaca perintah dari *user*.
2) *System Software*
   System Software berfungsi untuk mengoperasikan hardware (CPU, *Communication Module*, dan GUI) agar mampu menghasilkan fungsi dasar sesuai yang diinginkan.

3) *Communication Module*
   Communication Module (modul komunikasi) berfungsi untuk menterjemahkan data yang dikirim dari *sub system* pada kendaraan ke CPU dan sebaliknya.

4) GUI (*Graphical User Interface*)
   Graphical User Interface berupa tampilan informasi yang telah diproses oleh CPU agar mampu dibaca oleh *user* (pengendara).

2.3. Penelitian Terdahulu

2.3.1. *On-board Charge and Discharge Management System for Electric-Vehicle Batteries*

J. Alzieu, dkk (1994) mengembangkan sebuah *on-board charge dan discharge management system* untuk baterai mobil listrik. *On-board management system* yang mereka kembangkan digunakan pada mobil listrik yang dikembangkan oleh Electricité de France

(EDF) yang dilengkapi dengan baterai *valve–regulated lead/acid* (VRLA).

On-board management system ini dipasang di dalam kendaraan untuk mengatur kondisi baterai selama aktif. Konsep dari *on-board* ini didasarkan pada:

1) *Data Acquisition*
Mengumpulkan berbagai data dari komponen–komponen baterai.
2) *Data Processing*
Memproses data untuk menghindari kegagalan, kesalahan penggunaan, memprediksi SoC, dll.
3) *Data Recording*
Merekam seluruh data yang telah dikumpulkan dan diproses.

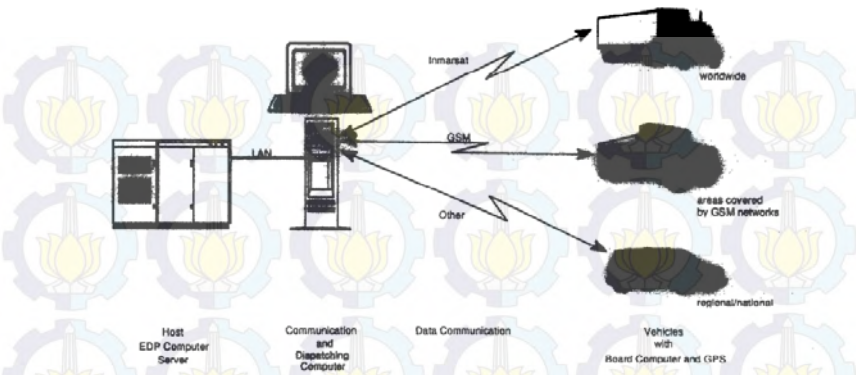
Fungsi utama yang didukung adalah: *battery life recording, charge monitoring*, *state-of-charge indication* (*gauge*), *maintenance*, *battery management during driving*.
Prototype dari *on-board management device* tersebut telah dikembangkan oleh IES (*Intelligent Electronic System*). Device yang telah dikembangkan tersebut mengumpulkan data dari seluruh *monoblock* battery. Tidak ada sensor yang cukup komplek yang digunakan. Device tersebut mengukur *individual voltage, current*, dan dua temperatur. Data dari setiap *monoblock* dikumpulkan selama fase *charging* dan *discharge*. Sebuah file permanen yang berisi ringkasan informasi diupdate setiap siklus kerja. Informasi temporer tentang kondisi masing – masing *monoblock* juga tersedia. Seluruh informasi atau data yang dikumpulkan dapat membantu dalam mendefinisikan peninggakatan kondisi dan menentukan penggantian *cell*.

2.3.2. *A Mobile Vehicle On-Board Computing and Communication System*

Berhard Sterzbach dan Wolfgang A. Halang (1996) telah mengembangkan sebuah *on-board system* pada kendaraan yang memiliki fungsi *computing* dan *communication system*. *System* tersebut dinamai DuO *vehicle tracking and fleet management*. Tujuan utama dari system yang mereka kembangkan adalah untuk mengumpulkan dan memproses informasi dari posisi aktual sebuah kendaraan yang bergerak secara *real-time* dengan menggunakan teknologi GPS dan GSM. Informasi yang telah dikumpulkan dan diproses tersebut kemudian akan di kirimkan ke *control center*.
Stuktur sistem dari DuO adalah sebagai berikut:

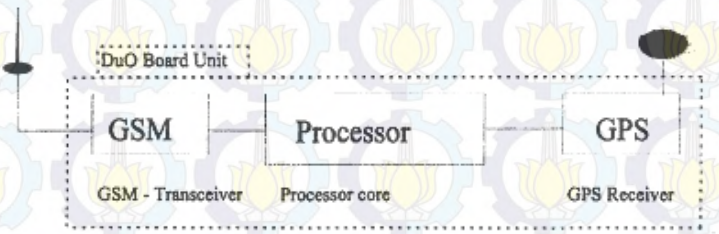
1) *Board Unit*
   Sebuah *on-board mobile computer* secara kontinyu mendapatkan dan memproses data posisi kendaraan.
2) *Control Centre*
   Satu atau lebih *computer* yang berfungsi menerima, menyimpan dan memvisualisasikan informasi dari *board unit*.
3) *Communication Network*
   Sebuah jaringan komunikasi *mobile* yang mentransmisikan data antara *board unit* dengan *control centre*.

Pada DuO *vehicle tracking system*, *board unit* menentukan lokasi fisik dan mentransmisikan informasi tersebut ke *control centre* pada interval yang teratur, dimana data tersebut akan disimpan di dalam *database* untuk memonitor pergerakan dari seluruh kendaraan.



Gambar 2.1 *Duo System Components (*Berhard Sterzbach dan Wolfgang A. Halang, 1996)

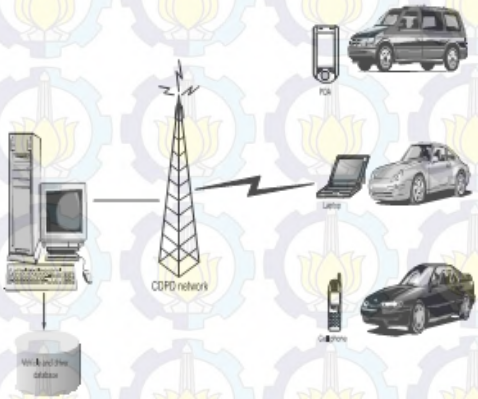
Mobile unit dari DuO terdiri dari modul untuk *geographic positioning system* (GPS), *data processing*, dan *mobile communication* yang terintegrasi dalam satu kotak. Modul posisi dan komunikasi tersambung dengan dengan modul *processor* melalui jaringan *serial*. *Operating power* didapatkan dari *power supply* kendaraan dengan menggunakan DC/DC *converter* berefisiensi tinggi.



Gambar 2.2 *Mobile Unit Block Diagram* (Berhard Sterzbach dan Wolfgang A. Halang, 1996)

2.3.3. *On-board Vehicle Data Stream Monitoring Using Mine-Fleet and Fast Resources Constrained Monitoring of Corelation Matrice*

Hillol Kargupta, dkk (2006) menulis sebuah paper yang membahas tentang permasalahan dalam memonitor aliran data kendaraan yang lebih difokuskan pada tugas *monitoring* yang membutuhkan komputasi dari korelasi matrik dengan menggunakan *lightweight on-board computing devices*. Dengan menggunakan MineFleet *system* / MineFleet *Real-Time* didesain untuk memonitor dan menggali aliran data kendaraan secara *real-time* yang kemudian ditampilkan melalui PDA/telepon selular atau *lightweight hardware* yang mirip dengan basis penggalian aliran data ataupun *remote desktop* berbasis modul *monitoring* yang tersambung dengan jaringan *wireless*.



Gambar 2.3 *Conceptual Depiction of The MineFleet System*
(Hillol KArgupta, dkk, 2006)

MineFleet *Real-Time system* terdiri dari empat komponen utama yaitu:

1) *On-board Hardware Module*
   Hardware interface untuk *on-board diagnostic* (OBD II) *data bus* yang digabung dengan *software* MineFleet *on-board* yang bekerja di sebuah PDA. Sebuah GPS juga dapat disambungkan dengan modul ini. Berbagai protokol komunikasi juga dapat digunakan di modul ini.

2) *On-board Data Stream Management and Mining Module*
   Modul ini menyediakan *run-time environment* untuk melakukan *on-board data analisis and management*. Modul *on-board monitor* mengatur aliran data dan memicu perintah ketika ditemui sebuah kondisi yang tidak umum. *On-board module* terhubung dengan *desktop* berbasis *remote* MineFleet *Control Centre* melalui jaringan *wireless*.

3) *Communication Network*

4) *Privacy Management Module*
   Sebuah jaringan komunikasi mobile yang mentransmisikan data antara *board unit* dengan *control centre*.

2.3.4. *Driver-Vehicle-Environment Monitoring for On-board Driver Support System: Lesson Learned from Design and Implementation*

Angelos Amditis, dkk (2010) mengembangkan DVE (*Driver-Vehicle-Environment*) yang ditujukan untuk mengkomputasi beberapa parameter yang dibutuhkan untuk menjalankan fungsi *display* adaptif berdasarkan deskripsi dari desain skenario AIDE dan kriteria yang relevan untuk adaptasi HMI untuk kondisi mengemudi tertentu secara real-time.

Modul monitoring DVE didesain untuk memproses kebutuhan lalu lintas/lingkungan sekitar, kebutuhan berkendara, kebingungan pengemudi, maksud pengemudi, dan ketidaknormalan fisik pengemudi; tujuan utamanya adalah untuk menentukan fungsi manakah yang akan dikomunikasikan dengan pengemudi dalam setiap keadaan.

DVE terdiri dari lima modul *monitoring* yang didesain dan dikembangkan untuk menjalankan fungsinya. Modul-modul tersebut telah terintegrasi dalam sebuah platform yang mampu menampilkan status yang disebut DVE *state*. DVE *state* meliputi :

1) *Input control information* (sudut *steering wheel*, posisi pedal, tombol-tombol, dll)
2) *Driver information* (pergerakan kepala/mata, kedipan pengemudi, dll).
3) *Environment and traffic information* (hambatan, jalanan, dll).
4) *Vehicle dynamic states* (kecepatan, percepatan, keolengan, dll).

Lima modul dari DVE adalah sebagai berikut:

1) *Traffic and Environment Risk Assessment* (TERA) *module*
   Berfungsi untuk mengestimasi keseluruhan resiko yang berhubungan dengan kondisi lalu-lintas dan kondisi lingkungan saat ini.
2) *Driver Characteristic* (DC) *module*
   Berfungsi untuk menyimpan dan mengkalkulasi parameter-parameter sesuai dengan profil pengemudi secara kontinyu.
3) *Driver Availability Estimator* (DAE) *module*
   Bertujuan untuk menilai tingkat kemampuan pengemudi untuk menerima dan memproses informasi.

4) *Driver State Degradation* (DSD) *module*
   Berfungsi untuk mendeteksi dan mendiagnosa secara *real-time* tingkat kesadaraan pengemudi terkait dengan rasa katuk maupun tertidur.
5) *Cockpit Activity Assessment* (CAA) *module*
   Berfungsi untuk memonitor tugas-tugas sekunder pengemudi (menyalakan radio berbicara dengan penumpang, dll).

2.3.5. Pengembangan IVI (*In Vehicle Infotainment*)

IVI adalah sebuah sistem yang terintegrasi didalam kendaraan yang menampilkan *entertainment* dan *infotainment*. IVI dapat ditemukan di beberapa mobil yang diproduksi oleh Ford, Toyota, KIA Motor, Cadillac, dan Fiat. IVI biasanya dilengkapi dengan teknologi *Bluetooth* dan/atau *Smartphone* untuk memudahkan pengendara mengkontrol sistem tersebut dengan suara, *touchscreen*, ataupun kontrol manual. Meskipun masing – masing sistem IVI dari setiap kendaraan berbeda–beda, secara umum IVI memiliki fungsi mengatur dan memutar musik, memberikan navigasi untuk berkendara, menampilkan hiburan seperti film, *game*, *social networking*, memberikan layanan komunikasi baik pesan SMS ataupun telepon, serta dapat mengakses internet untuk mengetahui situasi lalu lintas, laporan pertandingan olahraga, dan ramalan cuaca.

2.3.5.1. Pengembangan IVI oleh General Motor

GM (General Motor) melansir melalui *website* nya bahwa beberapa mobil yang dikembangkan oleh GM menawarkan teknologi yang canggih dan layanan hiburan. Pada mobil – mobil Chevrolet sistem ini dikenal dengan nama MyLink. Buick dan GMC menawarkan teknologi tersebut dengan nama IntelliLink. Kemudian pada Cadillac, sistem tersebut dijuluki CUE (Cadillac User Experience). Sistem–sistem tersebut telah mencakup berbagai fungsi termasuk AM/FM/XM *tuners*, CD *player* dengan MP3

playback, *auxiliary inputs*, dan USB 2 *input*. Tampilan yang digunakan adalah layar warna 7 in dengan resolusi tinggi dengan fitur *touchscreen* yang diklaim lebih mudah digunakan, lebih informatif, dan atraktif. Selain itu juga terdapat fitur komunikasi berupa pesan teks dan *streaming stereo audio* dari *smartphone* via koneksi *wireless* yang juga terintegrasi dengan *Bluetooth.*

2.3.5.2. Pengembangan IVI oleh NVIDIA

NVIDIA meluncurkan IVI sistem dengan menggunakan NVIDIA Tegra processors. Dengan Tegra K1 SoC, integrasi dari 4 CPU *cores*, dan penyimpanan *battery* 5^{th} *core*, dan juga sudah termasuk *video processor*, *image processor*, dan *audio processor* seluruhnya terpaket menjadi satu paket kecil dengan energy yang efisien. IVI milik NVIDIA memiliki fitur yang memudahkan pengendara dalam berkendara, tampilan yang mudah dibaca, tampilan *audio premium*, sistem navigasi 3D yang intuitive, pengenal suara, dan kontrol kokpit yang interaktif.

2.3.5.3. Pengembangan IVI oleh Mentor Graphics

Mentor Graphics telah meluncurkan IVI sistem yang berbasis Android dan Linux. Mentor Graphics menawarkan sistem infotainmen pada kendaraan yang berbeda dengan menggunakan GENIVI yang dilengkapi dengan *Automotive Technology Platform* (ATP) dari Mentor Embedded. Fokus dalam pengembangan fitur nilai tambah dengan mengurangi upaya integrasi ketika memanfaatkan platform berbasis teknologi Linux, software, dan servis dari Mentor Embedded. Secara detail sistem ini memiliki keunggulan sebagai berikut, telah menggunakan GENIVI dengan ATP.

2.4. *Individual System* pada Mobil Listrik Ezzy ITS

Mobil Listrik Ezzy ITS memiliki beberapa individual system. Pada Ezzy ITS terdapat 3 (tiga) individual system yang dapat dikategorikan sebagai sistem-sistem utama. 3 (tiga) individual system tersebut merupakan sistem-sistem yang menunjang fungsi utama Ezzy ITS sebagai mobil listrik. Saat ini ketiga sistem tersebut belum terintegrasikan. Adapun ketiga sistem tersebut adalah sebagai berikut.

2.4.1. Sistem Penggerak

Sistem penggerak pada Ezzy ITS terdiri dari 2 (dua) komponen utama yaitu *electric motor* dan *controller*. Pada Ezzy ITS, *electric motor* yang digunakan adalah PM 30 yang sudah dilengkapi bersama *controller*. Sistem penggerak pada Mobil Listrik Ezzy ITS tidak menyediakan protokol komunikasi. Oleh karena itu diperlukan penambahan sensor nantinya agar status dari motor dapat dimonitor.

2.4.2. Sistem Penyimpanan Energi

Sistem penyimpanan energi pada Ezzy ITS menggunakan baterai berjenis Lithium Polymer ($LiFePO_4$) yang dimanage dengan menggunakan Orion BMS (*battery management system*).

2.4.2.1. Baterai $LiFePO_4$

Baterai $LiFePO_4$ adalah baterai yang aman untuk digunakan pada mobil listrik. Selain itu baterai ini memiliki harga yang terjangkau dengan spesifikasi yang cukup baik. Karakteristik dari baterai ini adalah memiliki arus yang besar, *power* yang besar, kapasitas besar, ringan, dan berukuran kecil.

2.4.2.2. Orion BMS

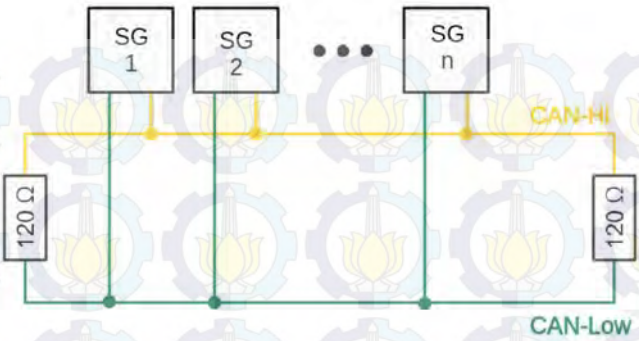
Orion BMS adalah sistem menejemen baterai untuk baterai lithium dengan berbagai fitur. Orion BMS didesain untuk memenejemen dan memproteksi baterai dari mobil listrik. Orion BMS menggunakan protokol komunikasi CAN dan *compatible* dengan OBD-II.
OBD-II adalah standar protokol komunikasi data untuk otomotif. OBD-II untuk komunikasi CAN distandarkan dalam ISO 15765 CAN (250 kBits/s atau 500 kBits/s).

2.4.3. Sistem Aksesoris

Sistem aksesoris pada mobil listrik meliputi sistem penerangan/lampu-lampu dan indikator keamanan seperti indikator *seatbelt*, pintu, dll. Sistem aksesoris ini mengeluarkan sinyal berupa *digital output* agar bisa dimonitor.

2.5. CAN BUS

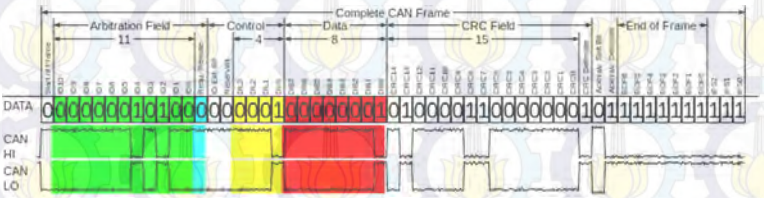
Pada pembahasan di atas telah disampaikan bahwa Orion BMS menyediakan protokol komunikasi berupa protokol CAN atau biasa disebut dengan CAN BUS. Oleh sebab itu penulis merasa perlu untuk menjelaskan lebih detail tentang CAN BUS. CAN BUS adalah protokol komunikasi yang umum digunakan dalam dunia Automotive, hamper seluruh kendaraan saat ini menggunakan protokol komunikasi CAN BUS. Protokol komunikasi tersebut digunakan sebagai jalur komunikasi antar komponen di dalam kendaraan.
CAN BUS digunakan pada kendaraan karena memiliki topologi yang sangat sederhana. Berikut adalah gambar topologi CAN BUS.



Gambar 2.4 Topologi CAN BUS (http://www.wikipedia.org)

Dari Gambar 2.4 dapat diketahui bahwa CAN BUS memiliki dua jalur transmisi data yaitu CAN High (CAN H) dan CAN Low (CAN L) yang menghubungkan titik-titik atau *subsystem*. Berbeda dengan protokol komunikasi yang lain dimana dua jalur dibedakan menjadi jalur penerima dan jalur pengirim. Pada CAN BUS jalur CAN H dan CAN L difungsikan sebagai keduanya. Sehingga nilai tegangan pada kedua jalur tersebut sama. Perbedaan sinyal pada kedua jalur tersebut hanya nilai tegangannya saja. Pada CAN H tegangan bernilai positif sedangkan pada CAN L nilai tegangan bernilai negatif.Kedua nilai tersebut akan dibandingkan oleh CAN *controller*. Hal tersebut membuat protokol CAN BUS menjadi protokol yang handal karena pengaruh intervensi tegangan tidak akan merusak isi informasi yang dikirimkan. Pada sinyal CAN BUS pesan atau informasi dikemas menjadi satu paket. Paket tersebut diidentifikasi dengan menggunakan ID dengan ukuran 11 bit untuk CAN Standard dan 29 bit untuk CAN Extended. Dengan ukuran 11 bit memungkinkan untuk membuat ID yang berbeda sebanyak 2048 ID. Kemudian setiap ID mampu membawa 8 paket data dengan ukuran 8 bit untuk masing-masing data. Dengan jumlah ID sebanyak 2048 ID dan 8 paket data masing-masing berukuran 8 bit, maka CAN BUS mampu mengirim dan menerima data dalam jumlah yang sangat banyak. Kemampuan tersebut juga

didukung dengan kecepatan transfer CAN BUS yang mencapai 1 Megabits/s.



Gambar 2.5 Format Data CAN BUS (http://www.wikipedia.org)

2.6. Gambaran Rencana Penelitian

Penelitian yang akan dilakukan adalah merancang dan membuat *integrated real-time monitoring system* (IRTMS) untuk mobil listrik Ezzy ITS yang berbeda dengan penelitian-penelitian yang sudah pernah dilakukan. Berdasarkan literatur-literatur di atas, dapat disimpulkan bahwa IRTMS yang dikembangkan sebelumnya lebih difokuskan sebagai fungsi *infotainment* dan *navigasi* serta *monitoring* posisi. Pada penelitian ini IRTMS akan difokuskan untuk tujuan mengintegrasikan sistem-sistem *monitoring internal* dari mobil listrik Ezzy ITS. IRTMS untuk mobil listrik Ezzy ITS ini akan diberi nama *In Vehicle Computer* (IVC). Adapun blok diagram dari sistem IVC adalah sebagai berikut.

Gambar 2.6 Blok Diagram *Input – Controller – Output* dari Sistem IVC

Berdasarkan gambar di atas dapat diketahui bahwa inputan untuk system IVC adalah data yang diperoleh dari system-system pada

ezzy ITS. Data–data tersebut kemudian diproses oleh sistem IVC sehingga menghasilkan output berupa tampilan GUI.

Sedangkan spesifikasi desain dari sistem IVC adalah sebagai berikut.

Hardware :
- Processor dengan kecepatan minimal 170 MHz
- Video Output (HDMI support)
- Audio Output
- USB 2.0 Port
- CAN Port
- Serial Port (UART)
- Storage

Software :
- Kompatible dengan kernel linux
- Cepat dan stabil
- Memiliki dokumentasi yang lengkap
- Mendukung Object Oriented Programming

GUI :
- Informasi yang ditampilkan jelas
- Desain intuitif

IVC untuk mobil listrik Ezzy ITS akan menggunakan *hardware* berupa Raspberry Pi Model B, STM32 F4 Discovery Board, dan TFT LCD. Sedangkan untuk *system software* yang akan dibuat dan digunakan pada IVC adalah *firmware* untuk STM32 F4 Discovery Board, IVC *Software* untuk Raspberry Pi Model B sekaligus GUI yang akan ditampilkan di TFT LCD. Adapun arsitektur dari IVC untuk mobil listrik Ezzy ITS adalah sebagai berikut.

Gambar 2.7 Arsitektur Sistem IVC

Diagram arsitektur di atas secara sederhana dapat menjelaskan alur kerja dari sistem IVC. Sistem penggerak, sistem penyimpanan energy, dan sistem aksesoris pada mobil listrik Ezzy ITS telah menyediakan data yang dapat dikumpulkan sebagai informasi performa dari masing-masing komponen pada sistem tersebut. Dalam hal ini STM32 F4 Discovery Board berfungsi untuk mengumpulkan data-data tersebut. Untuk dapat menjalankan tugas tersebut diperlukan sebuah *firmware* yang ditanamkan pada STM32 F4 Discovery Board tersebut. Pada tugas akhir ini nanti nya *firmware* akan dibuat dengan menggunakan bahasa C++. *Firmware* tersebut berfungsi untuk mengaktifkan pin CAN-H, CAN-L, *Digital Input*, dan UART (TX-RX) pada STM32 F4 Discovery Board. Selain itu, *firmware* tersebut juga harus mampu menterjemahkan tipe data CAN dan tipe data digital output menjadi tipe data *serial* yang dapat diakses oleh Raspberry Pi Model B melalui pin UART (TX – RX). Tipe data *serial* yang telah disediakan oleh STM32 F4 Discovery Board selanjutnya akan di akses oleh Raspberry Pi Model B dengan bantuan *software* IVC

yang terinstal di dalamnya. *Software* IVC ini juga akan dibuat pada tugas akhir ini dan dikembangkan menggunakan *software* QT Enterprise yang mendukung pengembangan Linux *Embedded Software*. *Software* IVC ini akan berjalan di atas kernel linux Boot to Qt. Software IVC juga berfungsi untuk mengolah data *serial* menjadi informasi yang dapat dimengerti oleh *user*/pengemudi berupa GUI (*Graphical User Interface*). GUI tersebut juga akan di desain pada tugas akhir ini dengan menggunakan QML (Qt *Modeling Language*) yang juga merupakan salah satu fitur yang dimiliki oleh Qt untuk mendesain aplikasi *user-interface*. GUI tersebut akan ditampilkan pada TFT LCD 7 in yang juga mendukung fitur *touchscreen*.

2.6.1. Sistem Penyimpanan Energi (Baterai $LiFePO_4$ dan Orion BMS)

Orion BMS digunakan untuk memanage performa dari baterai $LiFePO_4$. Orion BMS telah dilengkapi dengan *utility software* yang berguna untuk melakukan konfigurasi *profile* managemen baterai. Dari software tersebut dapat diketahui bahwa Orion BMS mengirimkan data CAN bus agar performa dari baterai dapat dimonitor.

Gambar 2.8 Tampilan Data CAN BUS di Utility Software Orion BMS

Terdapat tiga ID data pada Orion BMS yang mewakili berbagai macam data performa dari Baterai yang dihandle oleh Orion BMS. Performa baterai yang dapat dimonitor adalah sebagai berikut.

Table 2.1 Informasi Status Baterai yang dapat dimonitor pada Orion BMS

1	State of Charge	13	Lowest Resistance
2	Depth of Discharge	14	12V Supply
3	Pack Health	15	Total Pack Cycles
4	Pack Voltage	16	Battery Current
5	Pack Sum Voltage	17	Battery Power
6	Pack Open Voltage	18	Charge Limit
7	Pack Resistance	19	Discharge Limit
8	Highest Cell	20	Heatsink Temp
9	Lowest Cell	21	Highest Temp
10	Highest Open Cell	22	Lowest Temp
11	Lowest Open Cell	23	Temperature 1 – 4
12	Highest Resistance	24	Cell Voltage 1 - n

2.6.2. Sistem Aksesoris

Sistem aksesoris pada mobil listrik Ezzy ITS dapat dimonitor sebagai sinyal digital. Sistem aksesoris yang dapat dimonitor dari Ezzy ITS adalah sebagai berikut.

Table 2.2 *Digital Signal* pada Mobil Listrik Ezzy ITS

1	Lampu Kota	4	Lampu Sein
2	Lampu Jauh	5	Indikator Rem Tangan
3	Lampu Dekat	6	Indikator Hazard

2.6.3. Raspberry Pi Model B

Raspberry Pi Model B adalah sebuah *single board computer* seukuran kartu kredit yang dikembangkan di Inggris oleh Raspberry Pi Foundation dengan tujuan mempromosikan pembelajaran tentang dasar komputer di sekolah–sekolah.

Raspberry Pi Model B memiliki sebuah Broadcom BCM2835 *system on chip* (SoC) yang memuat ARM1176JZF-S 700 MHz *processor*, VideoCore IV GPU, dan 512 MB RAM. Raspberry Pi tidak dilengkapi dengan *hard disk* atau *solid state drive*, tetapi menggunakan sebuah SD *card* untuk booting sekaligus media penyimpanan data.

Table 2.3 Spesifikasi Raspberry Pi Model B

Raspberry Pi	**Model B**
SoC	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM, and single USB port)
CPU	700 MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set
GPU	Broadcom Video Core IV @ 250 MHz OpenGL, ES 2.0 (24 GFLOPS) MPEG-2 and VC-1 (with license), 1080p30 h.264/MPEG-4 AVC high profile decoder and encoder
Memory (SDRAM)	512 MB (shared with GPU)
USB 2.0 Ports	2 (via the built in integrated 3-port USB hub)
Video Input	A CSI input connector allows for the connection of a RPF designed camera module
Video Output	Composite RCA (PAL and NTSC), HDMI (rev 1.3 & 1.4), raw LCD Panels via DSI 14 HDMI resolution from 640x350 to 1920x1200 plus various PAL and NTSC standards
Audio Output	3.5 mm jack, HDMI, and, as of revision 2 boards, I^2S audio (also potentially for audio input)
Onboard Storage	SD / MMC /SDIO card slot (3.3V card power support only)

Onboard Network	10/100Mbps Ethernet (8P8C) USB adapter on the third port of the USB hub
Low-Level Peripherals	8 x GPIO, UART, I^2C bus, SPI bus with two chip selects, I^2S audio +3.3V, +5V, ground
Power Ratings	700 mA (3.5 W)
Power Source	5 Volt via MicroUSB or GPIO header
Size	85.60 mm x 53.98 mm (3.370 in x 2.125 in)
Weight	45 g (1.6 oz)
Operating System	Arch Linux ARM, Debian GNU/Linux, Gentoo, Fedora, FreeBSD, NetBSD, Plan 9, Raspbian OS, RISC OS, Slackware Linux

2.6.4. STM32 F3 ARM 32-bit Cortex™ Microcontroller

STM32 seri F3 adalah grup pertama dari microcontroller STM32 yang berbasis ARM Cortex-M4F core. Spesifikasi dari seri F4 ini adalah sebagai berikut:

- Core :
  ARM Cortex-M4F core at a maximum clock rate of 72 MHz.
- Memory :
  Static RAM consist of up to 192 KB general purpose, 64 KB core coupled memory (CCM), 4 KB battery backed, 80 bytes battery-backed with tamper-detection erase.
  Flash consist of 512 / 1024 / 2048 KB general purpose, 30 KB system boot, 512 bytes one-time programmable (OTP), 16 option bytes.
  Each chip has a factory-programmed 96-bit unique device identifier number.
- Peripherals :
  Common peripherals included in all IC packages are USB 2.0 OTG HS and FS, two CAN 2.0B, one SPI + two SPI or full duplex I^2S, three I^2C four USART, two UART, SDIO for SD/MMC cards, twelve 16-bit timers, two 32-bit timers,

two watchdog timers, temperature sensor, 16 or 24 channels into three ADCs, two DACs, 51 tp 140 GPIOs, sixteen DMA, improved real-time clock (RTC), cyclic redundancy check (CRC) engine, random number generator (RNG) engine, larger IC packages add 8/16-bit external memory bus capabilities.
The STM32F4x7 models add ethernet MAC and camera interface.
The STM32F41x/43x models add a cryptographic processor for DES / TDES / AES, and a hash processor for SHA-1 and MD5.
The STM32F4x9 models add a LCD-TFT controller.

- Oscillators consists of internal (16 MHz, 32 kHz), optional external (4 to 26 MHz, 32.768 to 1000 kHz).
- IC Packages : WLCSP64, LQFP64, LQFP144, LQFP176, UFBGA176.
- Operating Voltage range is 1.8 to 3.6 volt.

2.6.5. Qt Enterprise

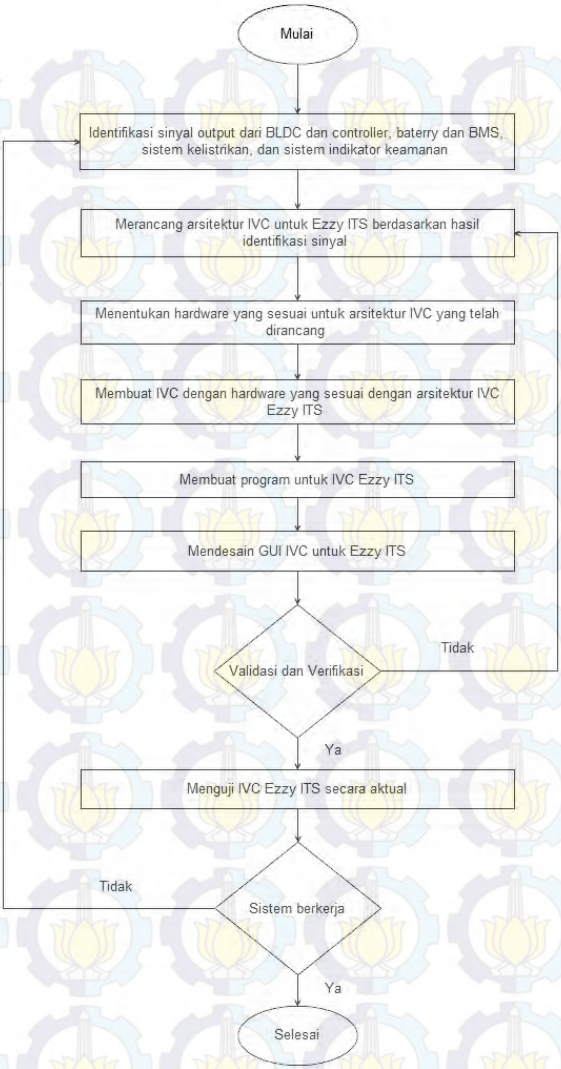
Qt adalah *cross-platforms native C++ libraries* yang sangat *powerful*, dengan UI (*user interface*) yang dapat dideklarasikan dan beberapa *tools* yang dimiliki Qt memungkinkan para pengembang untuk menggunakan satu kode dasar untuk ditargetkan pada seluruh *desktop*, *embedded*, *mobile* dan *real-time operating system* yang relevan. Qt memberikan kebebasan kepada pengembang untuk menghemat waktu pengembangan, menambahkan efisiensi dan mempersingkat waktu untuk pemasaran.
Qt Enterprise menyediakan segala kenyamanan dari penggunaan secara profesional untuk membuat aliran pekerjaan dari pengembangan menjadi selancar mungkin. Selain itu Qt Creator Enterprise yang juga tergolong sebagai *cross-platform Intergrated Development Environment* (IDE) menyediakan *tool* yang membantu produkstivitas dalam mendesain UI.

(Halaman ini sengaja dikosongkan)

BAB III
METODOLOGI PENELITIAN

Sebagai suatu proses yang terstruktur, dalam pelaksanaannya penelitian dan perancangan IVC untuk mobil listrik Ezzy ITS dilakukan dengan langkah-langkah sistematis. Metode dan langkah kerja dalam penelitian ini dijelaskan sebagai berikut.

3.1. Flowchart Perancangan IVC untuk Mobil Listrik Ezzy ITS



Gambar 3.1 Flowchart Perancangan Sistem IVC untuk Mobil Listrik Ezzy ITS

Flowchart pada Gambar 3.1 dapat dijelaskan sebagai berikut:

1) Identifikasi sinyal-sinyal output dari *electric motor* dan *controller*, *battery* dan BMS, serta sistem kelistrikan lampu dan sistem indikator keamanan pengemudi bertujuan untuk menentukan jumlah sinyal output dan jenis sinyal output. Hasil identifikasi tersebut akan digunakan untuk merancang arsitektur IVC untuk mobil listrik Ezzy ITS, menentukan *hardware*, dan *software* yang akan digunakan.

2) Setelah diketahui jumlah sinyal output dan jenisnya maka kita dapat merancang arsitektur IVC untuk mobil listrik Ezzy ITS. *Scope* perancangan adalah mendesain aliran kerja proses IVC, menentukan protokol komunikasi data yang sesuai, menentukan spesifikasi minimum *hardware* yang sesuai.

3) Pada tahap ini adalah menentukan *hardware* untuk IVC Ezzy ITS berdasarkan spesifikasi minimum yang telah ditentukan pada tahap sebelumnya.

4) Setelah *hardware* ditentukan dan didapatkan, maka tahap selanjutnya adalah membuat IVC sesuai dengan arsitektur yang telah dirancang sebelumnya.

5) Tahap selanjutnya adalah membuat *software* yang dapat dijalankan pada *hardware* yang ada dan mampu menjalankan perintah sesuai dengan arsitektur IVC.

6) Selanjutnya adalah mendesain GUI untuk menampilkan informasi berdasarkan data hasil olahan IVC agar mampu dibaca dan dimengerti oleh pengendara atau pengemudi.

7) Pada tahap ini IVC untuk mobil listrik Ezzy ITS dapat dikatakan sudah hampir selesai, maka diperlukan validasi dan verifikasi. Validasi dan verifikasi yang dilakukan adalah

dengan mensimulasikan IVC Ezzy ITS dengan menggunakan sinyal generator dan melihat apakah IVC sudah bekerja sesuai fungsinya. Jika belum maka perlu dilakukan peninjuan ulang terhadap arsitektur IVC. Jika sudah, maka dapat dilanjutkan ke tahap berikutnya.

8) Tahap terakhir adalah melakukan uji coba langsung terhadap IVC dengan menjalankan IVC langsung pada mobil listrik Ezzy ITS untuk melihat performa aktual dari IVC.

BAB IV
ANALISA DAN PEMBAHASAN

4.1. Identifikasi Sinyal Output dari Sistem Mobil Listrik Ezzy ITS

Identifikasi sinyal output dari sistem yang terdapat pada Mobil Listrik Ezzy ITS dilakukan untuk mengetahui jenis dan jumlah sinyal yang akan diproses oleh IVC sehingga dapat ditampilkan kepada pengemudi. Pada Mobil Listrik Ezzy ITS terdapat 3 jenis sinyal output yang berbeda yaitu sinyal output dalam bentuk data CAN BUS dari sistem baterai dan BMS, sinyal output dalam bentuk pulsa dari sistem BLDC dan *controller*, dan sinyal output dalam bentuk *digital* dari sistem aksesoris.

4.1.1. Sinyal CAN BUS dari Orion BMS

Pada Mobil Listrik Ezzy ITS baterai yang digunakan adalah baterai dengan jenis $LiFePO_4$ yang diatur dan dikelola oleh menggunakan Orion BMS. Orion BMS telah menyediakan protokol komunikasi dengan menggunakan CAN BUS. Sehingga status dari baterai $LiFePO_4$ pada Mobil Listrik Ezzy ITS dapat dimonitor melalui protokol komunikasi CAN BUS. Berikut adalah data-data hasil identifikasi data CAN BUS dari Orion BMS.

Gambar 4.1 Tampilan CAN BUS *Messages* pada Orion BMS Utility

Berdasarkan gambar 4.1 dapat diketahui bahwa terdapat 7 (tujuh) pesan yang berisi beberapa informasi dari baterai $LiFePO_4$ yang terpasang di Mobil Listrik Ezzy ITS. Dari ketujuh pesan tersebut dapat dipilih beberapa informasi yang dibutuhkan untuk ditampilkan kepada pengemudi. Berikut adalah table berisi data CAN BUS yang akan diproses untuk ditampilkan kepada pengemudi.

Table 4.1 Data CAN BUS yang akan dimonitor pada Sistem IVC

No.	ID	Data	Keterangan
1	1B	Data0	12 volt supply
2	2B	Data0	Average Temperature
3	3B	Data0&Data1	Pack Current
		Data2&Data3	Pack Voltage
4	3C	Data0&Data1	Average Voltage
5	5B	Data0&Data1	Lowest Cell Voltage
6	6B2	Data1	State of Charge

Ketujuh pesan di atas dipilih untuk ditampilkan kepada pengemudi dikarenakan ketujuh pesan tersebut merupakan pesan penting yang

harus diketahui oleh pengemudi secara *real-time*. Nilai 12 volt supply cukup penting untuk diketahui karena 12 volt supply merupakan power utama Orion BMS sehingga performa Orion BMS dipengaruhi oleh nilai dari 12 volt supply. *Average Temperature* (temperatur rata-rata baterai) juga penting untuk diketahui oleh pengemudi karena temperatur dapat mengindikasikan keamanan baterai. Selanjutnya nilai *Pack Current* (arus *discharge* pada baterai) merupakan informasi penting karena apabila arus terlalu besar akan menyebabkan baterai rusak. Kemudian *Pack Voltage* (nilai tegangan total baterai), nilai tegangan total penting untuk diketahui karena berdasarkan informasi tersebut pengemudi dapat memprediksi kapasitas baterai. *Average Voltage* (tegangan rata-rata *cell*) juga penting untuk diketahui karena merupakan indikator kesehatan tiap *cell*. *Lowest Cell Voltage* (nilai tegangan terendah) juga penting untuk diketahui karena apabila nilai tersebut terlalu kecil dapat mengakibatkan *cell* tersebut rusak. *State of Charge* (kapasitas baterai) nilai ini merupakan nilai terpenting karena mengindikasikan kapasitas baterai secara *real-time*.

4.1.2. Sinyal Pulse dari Magnetic Pulser

Pada Mobil Listrik Ezzy ITS BLDC dan kontroler yang digunakan tidak menyediakan protokol komunikasi sama sekali maka diperlukan penambahan sensor untuk membaca RPM motor dan kecepatan kendaraan.Sensor RPM digunakan untuk membaca rpm motor dan kecepatan kendaraan. Sensor RPM yang digunakan pada Mobil Listrik Ezzy ITS adalah magnetic pulser. Dari magnetic pulser tersebut akan didapatkan sinyal pulsa yang dapat merepresentasikan jumlah putaran motor persatuan waktu.

4.1.3. Sinyal Digital dari Sistem Aksesoris

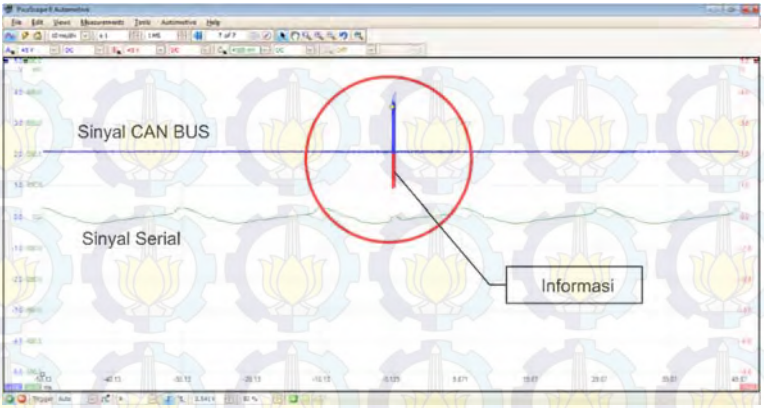
Pada Mobil Listrik Ezzy ITS yang dimaksud dengan sistem aksesoris adalah lampu-lampu dan *hand brake*. Sinyal-sinyal dari sistem aksesoris ini berupa sinyal digital yang menunjukkan

kondisi *on* dan *off*. Berikut adalah daftar sinyal-sinyal pada sistem aksesoris.

Table 4.2 Sinyal Indikator Sistem Aksesoris

No.	**Sinyal**
1	Idikator Lampu Sein
2	Indikator Hazard
3	Indikator Lampu Kota
4	Indikator Lampu Dekat
5	Indikator Lampu Jauh
6	Indikator *Hand Brake*

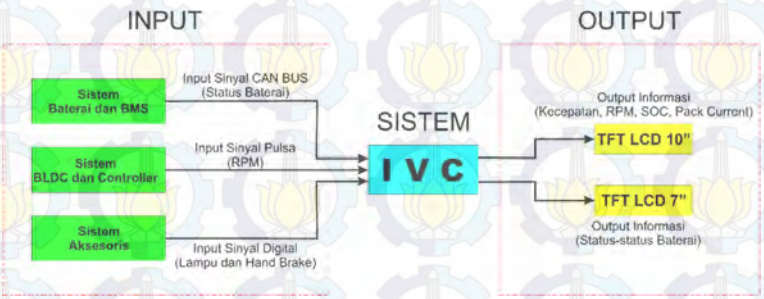
Setelah dilakukan identifikasi terhadap sinyal output dari sistem Mobil Listrik Ezzy ITS maka diputuskan untuk menterjemahkan ketiga sinyal tersebut menjadi 1 jenis sinyal agar proses monitoring menjadi lebih mudah dan efisien. Jenis sinyal yang dipilih adalah sinyal serial. Sinyal serial dipilih karena mudah digunakan dan merupakan protokol komunikasi yang umum. Selain itu sinyal serial juga mampu untuk meneruskan informasi dari sinyal CAN BUS yang memiliki kecepatan cukup tinggi. Hal tersebut ditunjukkan dari hasil simulasi konversi sinyal CAN BUS menjadi sinyal serial.



Gambar 4.2 Konversi Sinyal CAN BUS menjadi Sinyal Serial

Dari Gambar 4.2 dapat dilihat bahwa pada data serial (warna hijau) terjadi perubahan nilai tegangan ketika terdapat informasi pada sinyal CAN BUS (warna biru dan merah). Hal tersebut menunjukkan bahwa sinyal serial mampu meneruskan informasi dari sinyal CAN BUS dengan baik.

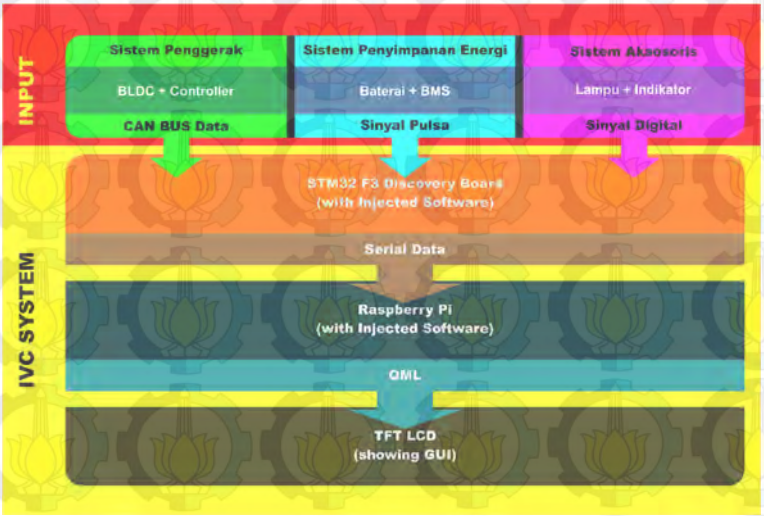
Berdasarkan hasil identifikasi sinyal-sinyal. Maka dapat digambarkan diagram input dan output sinyal pada sistem IVC nantinya adalah sebagai berikut.



Gambar 4.3 Diagram Sistem IVC

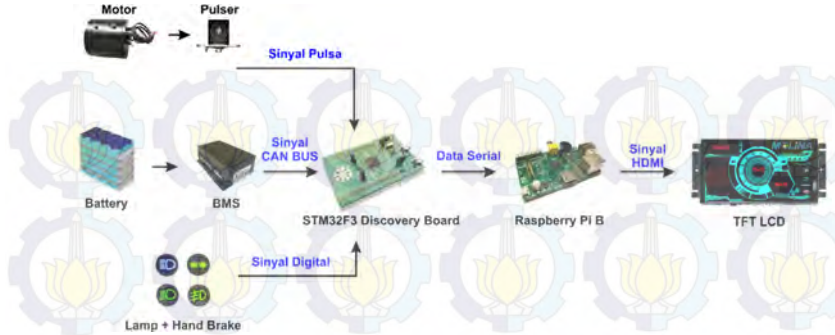
4.2. Perancangan Arsitektur IVC untuk Mobil Listrik Ezzy ITS

Perancangan Arsitektur IVC didasarkan pada fungsi utama IVC yaitu sebagai sistem *monitoring* untuk Mobil Listrik Ezzy ITS. Sedangkan dari proses identifikasi sinyal output pada Mobil Listrik Ezzy ITS telah diketahui bahwa terdapat 3 jenis sinyal output. Maka arsitektur IVC tentunya harus sesuai dengan kondisi tersebut. arsitektur IVC diharapkan mampu untuk mengelola ketiga jenis sinyal output tersebut untuk diterjemahkan menjadi 1 jenis sinyal agar proses *monitoring* menjadi lebih mudah. Jenis sinyal yang dipilih sebagai sinyal output dari Sistem IVC adalah sinyal dalam bentuk data serial. Berikut adalah Arsitektur IVC yang dirancang berdasarkan kebutuhan dan fungsi dari Sistem IVC



Gambar 4.4 Arsitektur IVC

Berdasarkan arsitektur pada Gambar 4.4 maka dapat digambarkan diagram fisik dari sistem IVC sebagai berikut.



Gambar 4.5 Sistem IVC

4.3. Perancangan Hardware IVC untuk Mobil Listrik Ezzy ITS

Perancangan Hardware IVC didasarkan pada Arsitektur IVC yang telah dirancang sebelumnya. Tahap awal dalam perancangan Hardware IVC adalah pemilihan komponen hardware yang akan dirangkai menjadi Hardware IVC. Tahap selanjutnya adalah Fabrikasi dan Assembly Hardware IVC.

4.3.1. Pemilihan Komponen Hardware IVC

Berdasarkan dari Arsitektur IVC maka dapat diketahui bahwa standar Hardware IVC harus memiliki kemampuan atau fungsi sebagai berikut.

1) Mampu mengelola data CAN BUS
2) Mampu mengelola data serial
3) Mampu mengelola sinyal pulsa
4) Mampu mengelola sinyal digital
5) Mampu menerjemahkan data serial menjadi informasi grafis yang ditampilkan melalui LCD

Berdasarkan dari ketentuan di atas maka komponen-komponen utama Hardware IVC yang dipilih adalah STM32F3 Discovery

Board sebagai *signal handler*, Raspberry Pi digunakan untuk mengirimakn sinyal agar informasi dapat ditampilkan di TFT, dan TFT LCD digunakan untuk menampilkan informasi dalam bentuk grafis.

4.3.1.1. STM32 F3 Discovery Board

STM32F3 Discovery Board dengan microcontroller STM32F3 digunakan sebagai signal handler pada Sistem IVC. Seluruh sinyal output dari sistem Mobil Listrik Ezzy ITS merupakan inputan bagi STM32F3. Sinyal-sinyal tersebut akan diterjemahkan oleh microcontroller tersebut menjadi satu paket data serial yang akan dikirimkan ke Raspberry Pi setiap ada permintaan dari Raspberry Pi.

4.3.1.2. Raspberry Pi

Raspberry Pi pada Sistem IVC digunakan untuk menampilkan informasi status sistem Mobil Listrik Ezzy ITS yang didapatkan dalam bentuk paket data serial. Informasi dari data serial tersebut selanjutnya akan diterjemahkan ke dalam bahasa QML sehingga dapat ditampilkan dalam bentuk informasi grafis pada TFT LCD.

4.3.1.3. TFT LCD

Display yang digunakan untuk menampilkan informasi status sistem pada Mobil Listrik Ezzy ITS dalam bentuk grafis adalah TFT LCD. TFT LCD dipilih karena memiliki kualitas resolusi yang tinggi dan memiliki dimensi yang cocok untuk digunakan di dalam mobil.

4.3.2. Fabrikasi dan Assembly Hardware IVC

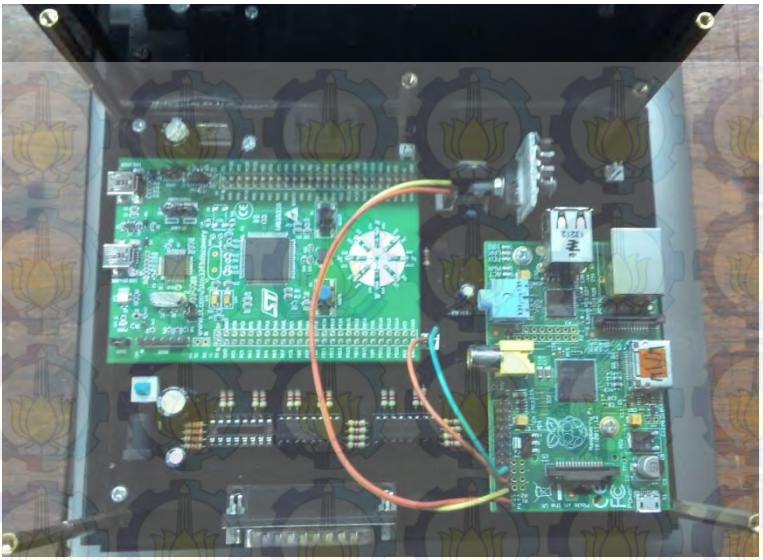
Setelah dapat ditentukan komponen hardware utama penyusun Sistem IVC selanjutnya dapat dilakukan proses fabrikasi dan assembly dari Hardware IVC.

4.3.2.1. Fabrikasi Hardware IVC

Proses fabrikasi Hardware IVC dalam hal ini adalah membuat mainboard untuk menyambungkan STM32F3 Discovery Board dengan Raspberry Pi. Dalam mainboard tersebut juga terdapat beberapa rangkain pendukung. Rangkaian pendukung tersebut antara lain adalah rangkain power regulator dan rangkaian signal conditioning. Rangkaian power regulator berguna untuk mengatur dan menstabilkan power yang digunakan untuk mengaktifkan STM32F3 Discovery Board dan Raspberry Pi. Sedangkan rangkaian signal conditioning digunakan untuk menstabilkan signal output dari sistem Mobil Listrik Ezzy ITS sebelum masuk ke microcontroller. Mainboard Hardware IVC didesain menggunakan software Altium Designer.

4.3.2.2. Assembly Hardware IVC

Setelah mainboard selesai dibuat maka proses selanjutnya adalah melakukan assembly komponen-komponen dari Hardware IVC Tersebut.



Gambar 4.6 Hardware IVC setelah Diassembly

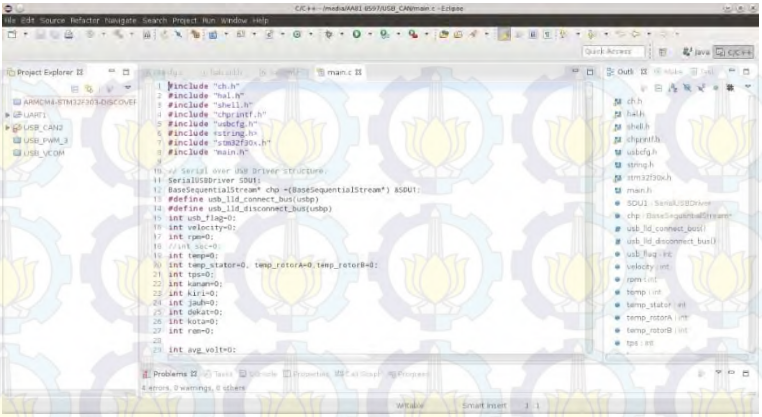
4.4. Perancangan Software IVC untuk Mobil Listrik Ezzy ITS

Perancangan Software IVC dilakukan dalam 2 tahap yaitu perancangan software untuk microcontroller STM32F3 dan perancangan software untuk Raspberry Pi.

4.4.1. Perancangan Software Microcontroller STM32 F3

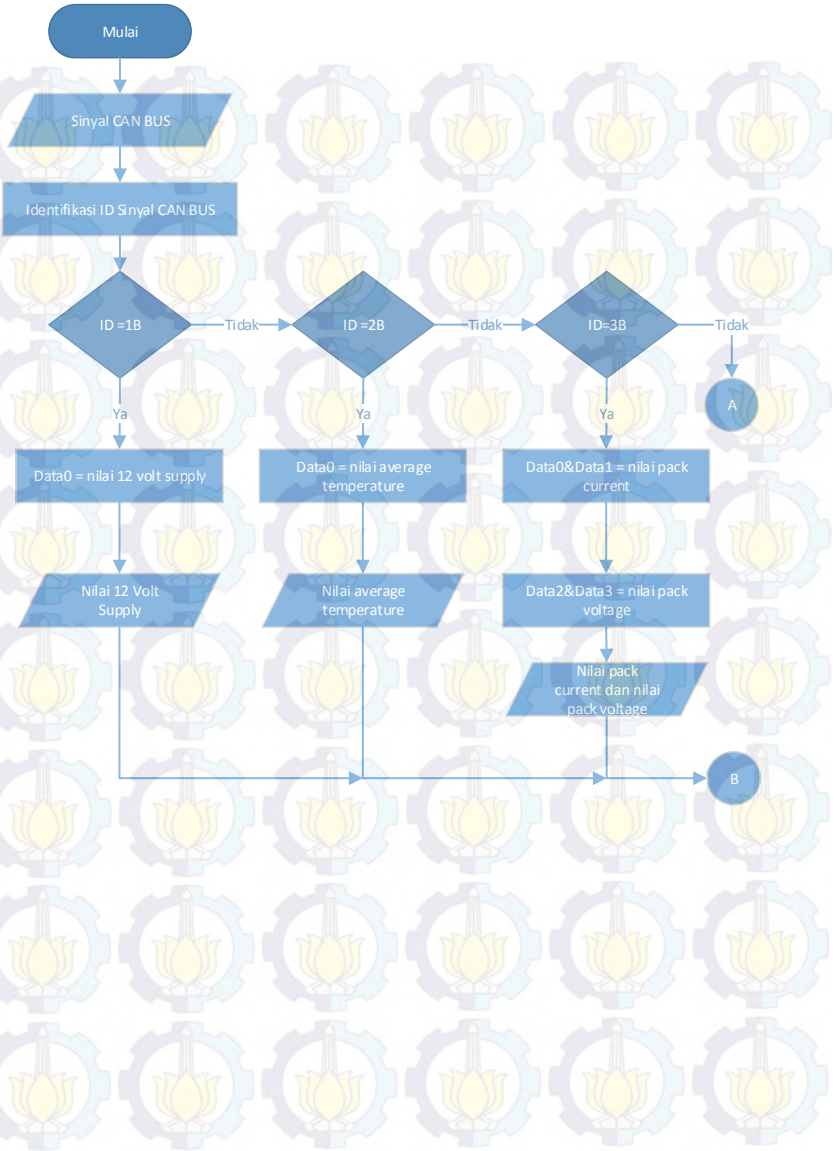
Software yang dirancang untuk microcontroller STM32F3 merupakan software signal handler yang berfungsi untuk mengelola sinyal-sinyal output dari sistem Mobil Listrik Ezzy ITS. Sinyal-sinyal tersebut selanjutnya akan dikirimkan ke Raspberry Pi dalam bentuk paket data serial. Paket data serial tersebut dikirimkan apabila ada permintaan dari Raspberry Pi.

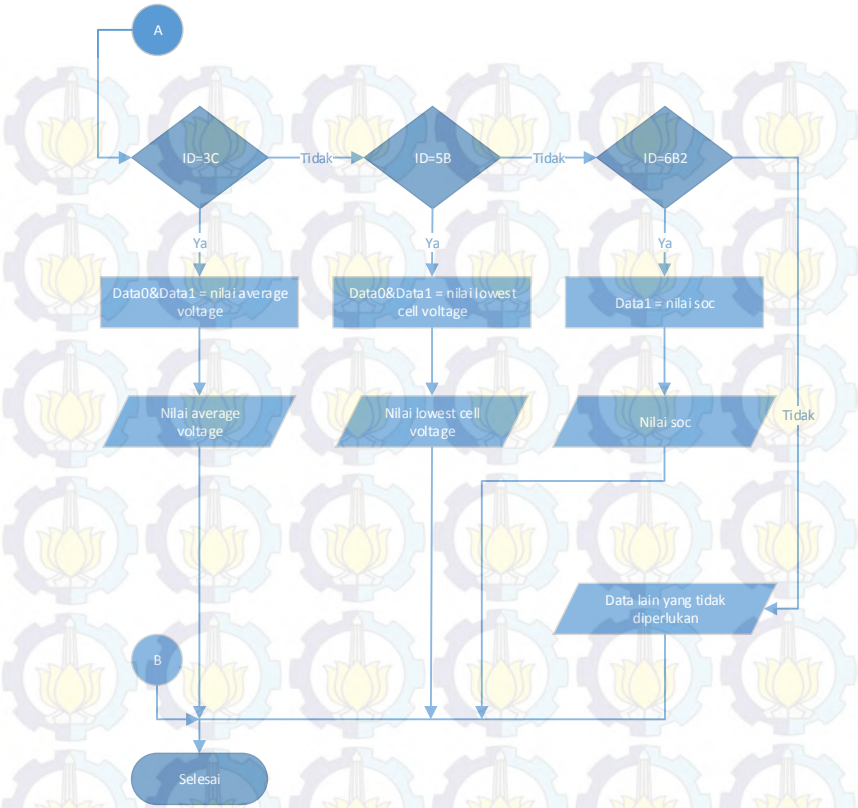
Software microcontroller STM32F3 ditulis menggunakan bahasa C dengan bantuan software Eclipse IDE.



Gambar 4.7 Tampilan Software Microcontroller STM32 F3 pada Eclipse IDE

Berikut adalah flowchart algoritma software microcontroller STM32F3.


Mulai
Sinyal CAN BUS
Identifikasi ID Sinyal CAN BUS
ID =1B
Tidak
ID =2B
Tidak
ID=3B
Tidak
A
Ya
Ya
Ya
Data0 = nilai 12 volt supply
Data0 = nilai average temperature
Data0&Data1 = nilai pack current
Nilai 12 Volt Supply
Nilai average temperature
Data2&Data3 = nilai pack voltage
Nilai pack current dan nilai pack voltage
B



Gambar 4.8 Flowchart Algoritma Program CAN BUS Signal Handler

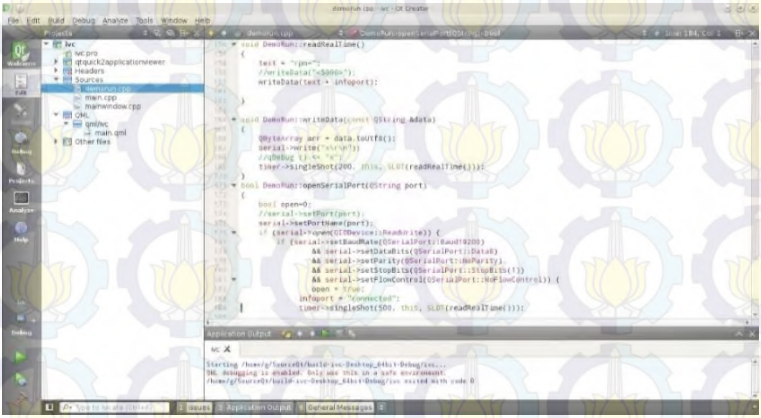
Gambar 4.9 Flowchart Algoritma Program Pulse Signal Handler

Gambar 4.10 Flowchart Algoritma Program Digital Signal Handler

4.4.2. Perancangan Software Raspberry Pi

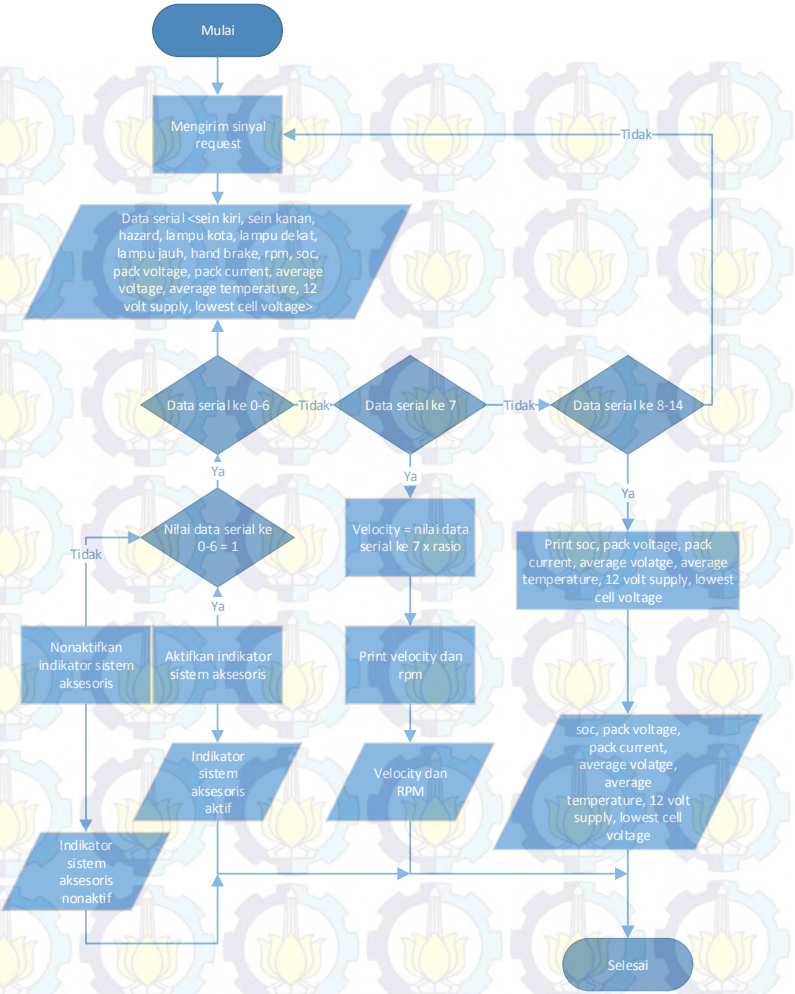
Software untuk Raspberry Pi merupakan software yang berfungsi untuk menampilkan informasi status sistem pada Mobil Listrik Ezzy ITS dalam bentuk informasi grafis di TFT LCD. Informasi yang diterima Raspberry Pi merupakan paket data serial yang dikirimkan oleh STM32F3 setelah Raspberry Pi mengirimkan sinyal request ke STM32F3.

Software Raspberry Pi ditulis dengan menggunakan bahasa C++ dan QML dengan bantuan software Qt.



Gambar 4.11 Tampilan Software Raspberry Pi pada Qt

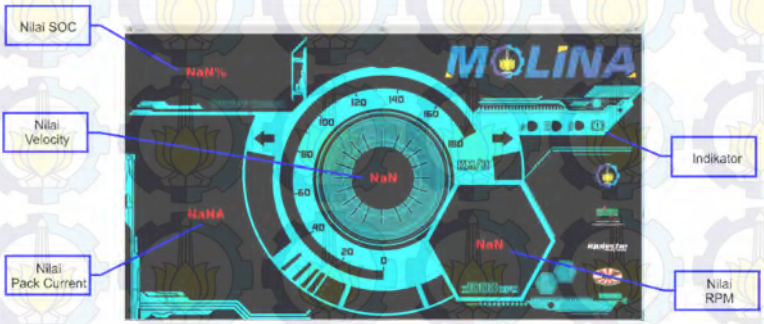
Berikut adalah flowchart algoritma dari Software Raspberry Pi.



Gambar 4.12 Flowchart Algoritma Program pada Raspberry Pi

4.5. Perancangan GUI IVC untuk Mobil Listrik Ezzy ITS

GUI IVC juga dirancang dengan menggunakan software Qt. Komponen GUI yang dirancang dengan menggunakan software Qt adalah komponen animasi yang memungkinkan informasi dapat berubah secara real-time di TFT LCD. Komponen animasi tersebut ditulis dengan menggunakan bahasa QML (Qt Modelling Language). Selain komponen animasi juga terdapat komponen statis dari GUI IVC. Komponen statis merupakan background yang berfungsi untuk memperindah tampilan sehingga informasi dapat ditampilkan dengan menarik. Komponen statis dirancang dengan bantuan software Corel Draw.



Gambar 4.13 Tampilan Rancangan GUI IVC untuk TFT LCD 10”

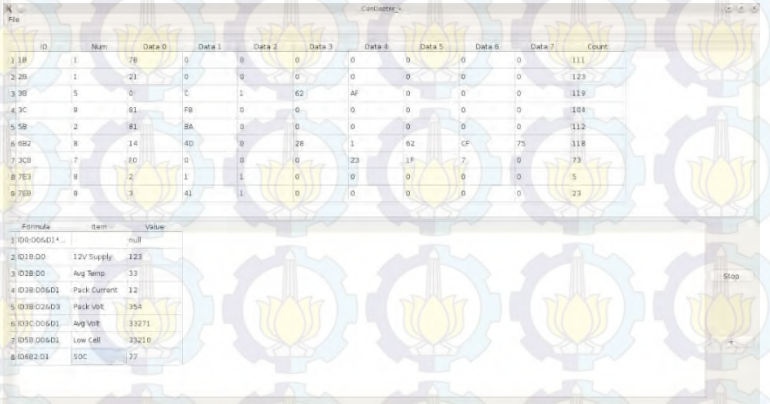
Gambar 4.14 Tampilan Rancangan GUI IVC untuk TFT LCD 7”

4.6. Proses Validasi dan Verifikasi Sistem IVC

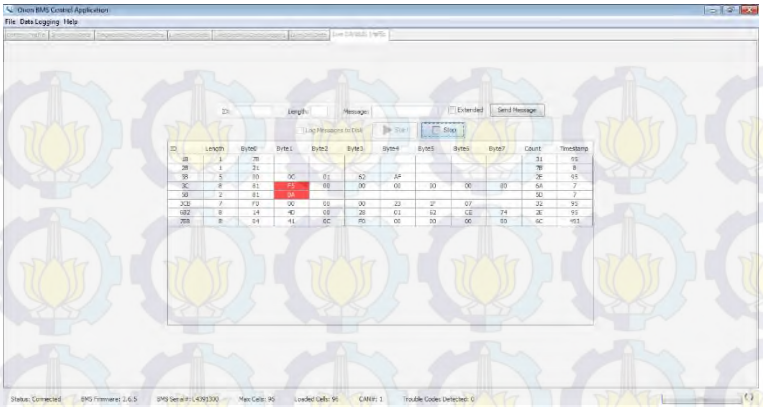
Proses validasi dan verifikasi dilakukan dalam dua tahap yaitu validasi dan verifikasi inputan sinyal untuk STM32F3 Discovery Board dan validasi dan verifikasi tampilan informasi yang dihasilkan oleh Raspberry Pi.

4.6.1. Validasi dan Verifikasi pada STM32 F3 Discovery Board

Proses validasi dan verifikasi sinyal input pada STM32F3 Discovery Board dilakukan dengan bantuan software pembaca data (CAN_Dapter) CAN BUS dan indicator led. Software pembaca data CAN BUS digunakan untuk mengetahui data CAN BUS yang diterima oleh STM32F3. Setelah data CAN BUS dapat diketahui selanjutnya adalah membandingkan nilai-nilai pada data tersebut dengan nilai-nilai yang terdapat pada Software resmi Orion BMS. Apabila nilai-nilai dari data CAN BUS telah sama maka data CAN BUS yang diterima oleh STM32F3 Discovery Board telah valid dan terverifikasi.



Gambar 4.15 Tampilan Data CAN BUS yang Diterima STM32 F3 Discovery Board



Gambar 4.16 Tampilan Data CAN BUS pada Software Orion BMS Utility

Berdasarkan Gambar 4.15 dan Gambar 4.16 dapat diketahui bahwa data CAN BUS yang diterima oleh STM32F3 Discovery Board telah sama dengan data CAN BUS yang ditampilkan di software Orion BMS Utility. Berikut adalah tabel perbandingan pembacaan data dari software CAN_Dapter dengan software Orion BMS Utility.

Table 4.3 Perbandingan Data CAN_Dapter dengan Orion BMS Utility

No.	ID	Data0		Data1		Data2		Data3	
1	1B	7B	7B						
2	2B	21	21						
3	3B	0	00	C	0C	1	01	62	62
4	3C	81	81	F8	F5				
5	5B	81	81	BA	BA				
6	6B2	14	14	4D	4D			28	28

CAN_Dapter

Orion BMS Utility

Berdasarkan Table 4.3 diatas dapat dilihat bahwa perbedaan nilai yang ditampilan pada software CAN_Dapter dan Orion BMS Utility hanya terdapat pada sinyal dengan ID 3C data ke 1. Hal tersebut dikarenakan terdapat jeda waktu pada saat pengambilan data. Tetapi perbedaan nilai tersebut tidak terlalu signifikan yaitu 33272 pada CAN_Dapter dan 33269 pada Orion BMS Utility atau hanya selisih 3 saja.

Sedangkan untuk validasi sinyal digital dan sinyal pulsa digunakan led pada STM32F3 Discovery Board sebagai indikator. Led akan aktif apabila terdapat inputan sinyal digital maupun sinyal pulsa.

4.6.2. Validasi dan Verifikasi pada Raspberry Pi

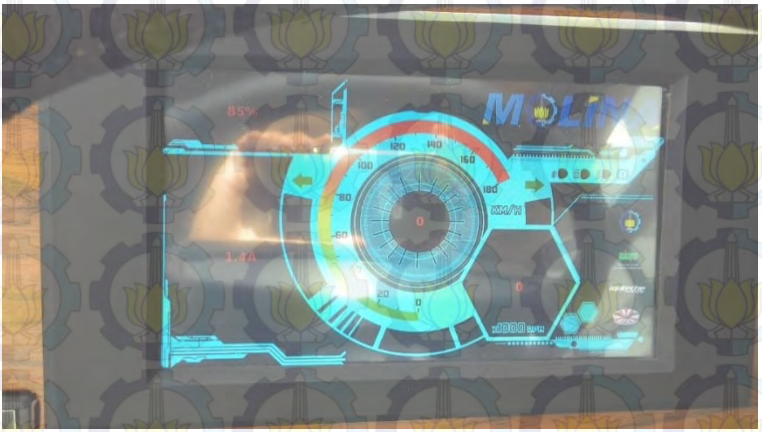
Selanjutnya proses validasi dan verifikasi tampilan informasi grafis yang dihasilkan oleh Raspberry Pi dilakukan dengan memberikan inputan data serial yang dihasilkan dengan bantuan software simulator. Dengan bantuan software simulator tersebut dapat diketahui apakah informasi grafis yang ditampilkan telah sesuai dengan nilai yang dikirimkan dalam bentuk paket data serial oleh software simulasi.

Gambar 4.17 Tampilan Software Simulator IVC

4.7. Proses Pengujian Sistem IVC

Pada tugas akhir ini pengujian Sistem IVC dilakukan pada Bis Listrik. Hal tersebut dikarenakan pada saat tugas akhir ini dikerjakan sistem dari Mobil Listrik Ezzy ITS sedang di bongkar untuk keperluan riset. Bis Listrik dipilih untuk digunakan menguji Sistem IVC karena Bis Listrik memiliki sistem yang sama dengan Mobil Listrik Ezzy ITS sehingga hasil pengujian dengan menggunakan Bis Listrik dapat digunakan sebagai indikator performa Sistem IVC.

Gambar 4.18 Tampilan Informasi Status Baterai saat Sistem IVC Diuji pada Bis Listrik



Gambar 4.19 Tampilan Informasi Utama saat Sistem IVC Diuji pada Bis Listrik

Berdasarkan dari Gambar 4.18 dan Gambar 4.19 dapat diketahui bahwa Sistem IVC telah mampu untuk menampilkan informasi dari status sistem pada Bis Listrik. Data CAN BUS yang disediakan oleh Orion BMS pada Bis Listrik telah ditampilkan sesuai dengan nilai aktual dan secara real time. Selanjutnya

indikator Sistem Aksesoris juga dapat ditampilkan pula dengan baik. Berikut adalah table hasil tampilan informasi dari sistem IVC dibandingkan dengan informasi dari software Orion BMS Utility (data CAN BUS).

Table 4.4 Perbandingan Sistem IVC dengan Orion BMS Utility

No.	Data	Sistem IVC	Orion BMS Utility
1	12 Volt Supply	12 V	12 V
2	Average Temperature	29°C	29°C
3	Pack Current	1.4 A	1.4 A
4	Pack Voltage	343 V	343 V
5	Average Voltage	3.3271 V	3.327 V
6	Lowest Cell Voltage	3.3195 V	3.319
7	State of Charge	85%	85%

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah perancangan dan pembuatan serta pengujian dari Sistem IVC, maka didapatkan kesimpulan sebagai berikut:

1) Arsitektur Sistem IVC telah berhasil dirancang dan dapat berfungsi untuk memonitoring status dari sistem penggerak, sistem penyimpanan energi, dan sistem aksesoris pada Mobil Listrik Ezzy ITS secara real-time.
2) Komponen hardware utama yang digunakan untuk sistem IVC adalah STM32F3 Discovery Board, Raspberry Pi, dan TFT LCD.
3) Software IVC telah berhasil dirancang dan dapat berfungsi untuk menyampaikan informasi dari sistem penggerak, sistem penyimpanan energi, dan sistem aksesoris di Mobil Listrik Ezzy ITS kepada pengemudi.
4) GUI IVC yang telah dirancang dapat menampilkan informasi dalam bentuk grafis dengan baik kepada pengemudi.
5) Secara keseluruhan sistem IVC telah berhasil dirancang dan dapat berfungsi sebagai sistem *monitoring* untuk Mobil Listri Ezzy ITS.

5.2. Saran

Sistem IVC yang dikembangkan masih dalam tahap prototype, sehingga masih perlu dilakukan penyempurnaan dengan menggantikan seluruh sinyal dengan sinyal CAN BUS. Karena pada beberapa kondisi sinyal non CAN BUS dapat terganggu. Selain itu dengan hanya menggunakan sinyal CAN BUS akan membuat sistem di Mobil Listrik menjadi lebih praktis dan efisien. Kedepan IVC juga perlu dikembangkan menjadi Mini Computer yang fungsinya tidak hanya sebagai sistem monitoring saja.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

[1] Alzieu, J., dkk. (1994). ”Development of an on-board charge and discharge management system for electric-vehicle batteries”. *Journal of Power Sources* 53 327-333.

[2] Amditis, A., dkk. (2010). “Driver-Vehicle-Environment monitoring for on-board driver support systems: Lesson learned from design and implementation”. *Applied Ergonomics* 41 225-235.

[3] Digia Oyj. 2013. Power. Beauty. Portability. Target Everything with Qt. http://qt.digia.com/ . 1 Maret 2014.

[4] General Motor. 2014. Innovation: Design & Technology. http://www.gm.com/vision/design_technology/in-vehicle_infotainment.html. 1 Maret 2014.

[5] Kargupta, H.., dkk. (2006). “On-board Vehicle Data Stream Monitoring Using Mine-Fleet and Fast Resource Constrained Monitoring of Corelation Matrices”. *New Generation Computing* 25 5-32.

[6] Mentor Graphics. 2014. Automotive Infotainment. http://www.mentor.com/embedded-software/automotive/in-vehicle-infotainment/ . 1 Maret 2014.

[7] NVIDIA Corporation. 2014. ENJOY THE RIDE WITH NVIDIA IN-VEHICLE INFOTAINMENT (IVI). http://www.nvidia.com/object/automotive-infotainment-navigation.html . 1 Maret 2014.

[8] Orion BMS. 2014. Battery Management System. http://www.orionbms.com/ . 20 Maret 2014.

[9] Sevcon. 2012. Gen 4 Size 8. http://www.sevcon.com/ac-controllers/gen-4-size-8.aspx . 20 Maret 2014.

[10] Sterzbach, B.., dkk. (1996). “A MOBILE VEHICLE ON-BOARD COMPUTING AND COMMUNICATION SYSTEM”. *Comput &Graphics*, Vol. 20, No 4, pp. 659-667, 1996

[11] STMicroelectronics. 2014. STM32F4DISCOVERY. http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF252419 . 1 Maret 2014.

[12] Wikipedia. 2014. Raspberry Pi. http://en.wikipedia.org/wiki/Raspberry_Pi . 1 Maret 2014.

[13] Wright, D.S. (1990). “Current on-board vehicle system”. *Transportation* 17:239-250, .

[14] Yasa Motors. 2013. YASA-750. http://www.yasamotors.com/products/yasa-750/ . 20 Maret 2014.

LAMPIRAN

LAMPIRAN 1 PROGRAM STM32 F3 Discovery Board

```
#include "ch.h"
#include "hal.h"
#include "shell.h"
#include "chprintf.h"
#include "usbcfg.h"
#include <string.h>
#include "stm32f30x.h"
#include "main.h"
//////
//Main program
int main(void) {
  Thread *shelltp = NULL;//deklarasi thread shell comand
  //inisialisasi HAL dan OS
  halInit();
  chSysInit();
     /*
   * Activates the serial driver 1 using the driver default
configuration.
   * PA9(TX) and PA10(RX) are routed to USART1.
   */
   icuStart(&ICUD1, &icucfg);
  palSetPadMode(GPIOE, 9, PAL_MODE_ALTERNATE(2));
  icuEnable(&ICUD1);
  chThdSleepMilliseconds(2000);
  sdStart(&SD1, NULL);
  sdStart(&SD2, NULL);
  static WORKING_AREA(waShellsd1, 2048);
  shellCreateStatic(&shell_sd1, waShellsd1, sizeof(waShellsd1),
NORMALPRIO+1);
  palSetPadMode(GPIOA, 9, PAL_MODE_ALTERNATE(7));
  palSetPadMode(GPIOA, 10, PAL_MODE_ALTERNATE(7));
```

```
  static WORKING_AREA(waShellsd2, 2048);
  shellCreateStatic(&shell_sd2, waShellsd2, sizeof(waShellsd2),
NORMALPRIO+2);
  palSetPadMode(GPIOA, 14, PAL_MODE_ALTERNATE(7));
  palSetPadMode(GPIOA, 15, PAL_MODE_ALTERNATE(7));
CAN_Config();
  //inisialisasi usb
  sduObjectInit(&SDU1);
  sduStart(&SDU1, &serusbcfg);
  //aktifkan usb
  usbDisconnectBus(serusbcfg.usbp);
  chThdSleepMilliseconds(200);
  usbStart(serusbcfg.usbp, &usbcfg);
  usbConnectBus(serusbcfg.usbp);
  //inisialisasi shell comand
  shellInit();
  chThdCreateStatic(waThread_calc, sizeof(waThread_calc),
NORMALPRIO+3, Thread_calc, NULL);
  while (TRUE)
  {
        //khusus USB Comand
    if (!shelltp && (SDU1.config->usbp->state ==
USB_ACTIVE))
        {
              shelltp = shellCreate(&shell_cfg1,
SHELL_WA_SIZE, NORMALPRIO);//mengaktifkan shell
comand
              usb_flag=1;
        }
    else if (chThdTerminated(shelltp))
        {
                    chThdRelease(shelltp);    /* Recovers
memory of the previous shell.   */
                    shelltp = NULL;          /* Triggers
spawning of a new shell.        */
```

```
      }
    ///~usb
     CAN_Transmit(CANx, &TxMessage);
     GPIO_E->ODR ^= (1<<13);
     chThdSleepMilliseconds(500);//delay looping main thread
     //rpm += 1;
    // if (rpm==255)rpm=0;
      ///
   }
  return 0;
}
  //chprintf((BaseSequentialStream *)&SDU1, "<%X,",
RxMessage.StdId);
  //chprintf((BaseSequentialStream *)&SDU1, "%d,",
RxMessage.DLC);
  chprintf(chp, "<%X,", RxMessage.StdId);
  chprintf(chp, "%d,", RxMessage.DLC);

  for(i=0;i<RxMessage.DLC ;i++)
          chprintf(chp, "%X,",RxMessage.Data[i]);
  if(RxMessage.DLC<8){
    for(i=0;i<(8-RxMessage.DLC) ;i++)
         chprintf(chp, "0,");
  }
  //chprintf((BaseSequentialStream *)&SDU1, ">,",
RxMessage.StdId);
}
CH_IRQ_HANDLER(STM32_CAN1_RX0_HANDLER) {

  CH_IRQ_PROLOGUE();
//GPIO_E->ODR ^= (1<<9);
  palTogglePad(GPIOE, GPIOE_LED9_BLUE);
  CAN_Receive(CAN1, CAN_FIFO0, &RawMessage);
   switch(RawMessage.StdId){
     case 0x1B:
```

LAMPIRAN 2 PROGRAM Raspberry Pi

```
#include "demorun.h"
#include <QtSerialPort/QSerialPort>
#include <QtCore/QtMath>
#include <QDebug>
DemoRun::DemoRun(QObject *parent) :
    QObject(parent)
{
    timer = NULL;
    timer = new QTimer(this);
    connect(timer, SIGNAL(timeout()), this, SLOT(timeout()));
    //timer->start(200);
    timerserial = NULL;
    timerserial = new QTimer(this);
    /*btn1 =  new QPushButton;
    connect(btn1, SIGNAL(clicked()), this,
SLOT(btn1_clicked()));
    btn2 =  new QPushButton;
    connect(btn2, SIGNAL(clicked()), this,
SLOT(btn2_clicked()));*/
    /*foreach (const QSerialPortInfo &info,
QSerialPortInfo::availablePorts()) {
        infoport.append(info.portName()+"\n\r");
        // Example use QSerialPort
        QSerialPort serial;
        serial.setPort(info);
        if (serial.open(QIODevice::ReadWrite))
            serial.close();
    }*/
    serial = new QSerialPort;
    connect(serial, SIGNAL(readyRead()), this,
SLOT(readData()));
    btn1_clicked();
}
int DemoRun::rpmvalue() const
```

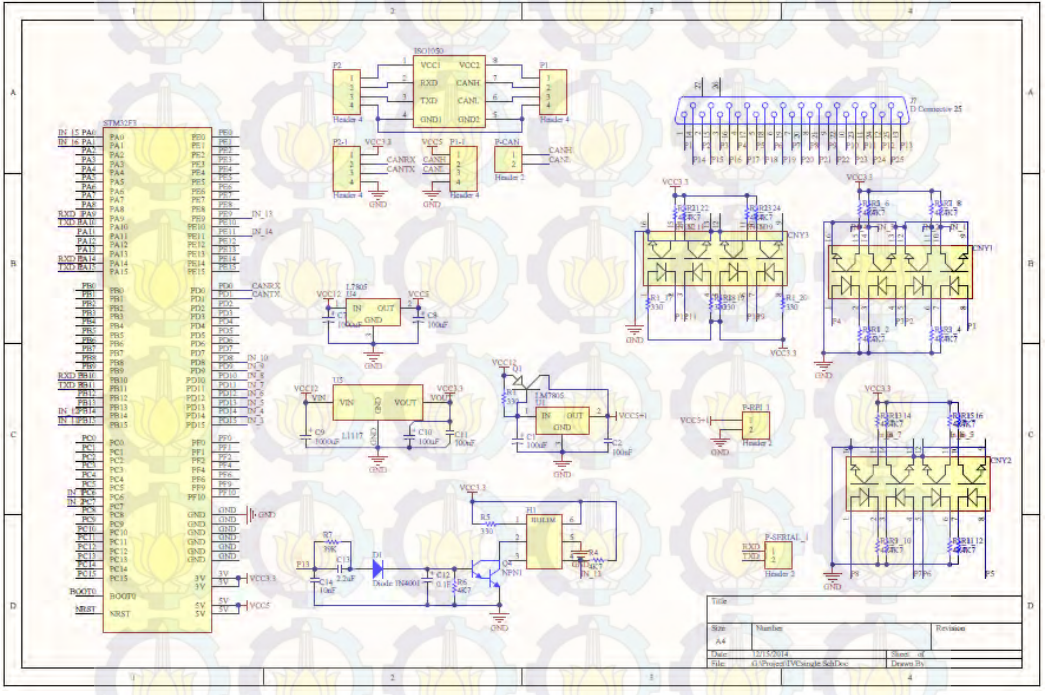
```
void DemoRun::readRealTime()
{
    text = "rpm=";
    //writeData("<5000>");
    writeData(text + infoport);
}
void DemoRun::writeData(const QString &data)
{
    QByteArray arr = data.toUtf8();
    serial->write("x\r\n");
    //qDebug () << "x";
    timer->singleShot(200, this, SLOT(readRealTime()));
}
bool DemoRun::openSerialPort(QString port)
{
    bool open=0;
    //serial->setPort(port);
    serial->setPortName(port);
    if (serial->open(QIODevice::ReadWrite)) {
        if (serial->setBaudRate(QSerialPort::Baud19200)
                && serial->setDataBits(QSerialPort::Data8)
                && serial->setParity(QSerialPort::NoParity)
                && serial->setStopBits(QSerialPort::StopBits(1))
                && serial-
>setFlowControl(QSerialPort::NoFlowControl)) {
                open = true;
            infoport = "connected";
            timer->singleShot(500, this, SLOT(readRealTime()));
        } else {
            open = false;
            serial->close();
        }
    }
    return open;
}
```

LAMPIRAN 3 PROGRAM QML pada Raspberry Pi

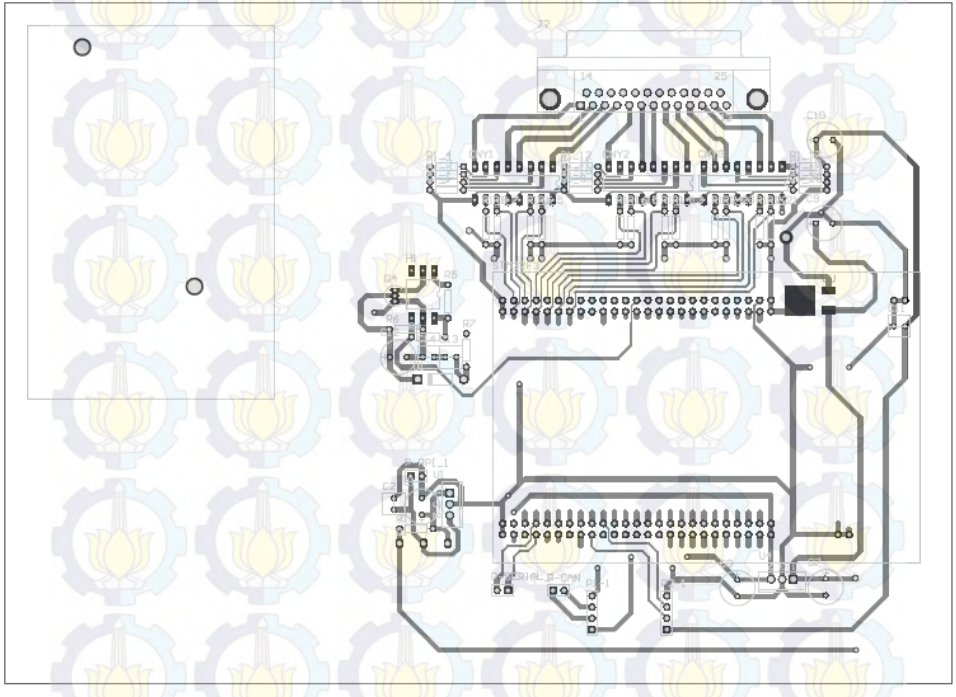
```
import QtQuick 2.0
import DemoRun 1.0
Rectangle {
   width: 1280
   height: 760
   color: "black"
   function timeChanged() {
      info = engine.author
   }
  Timer {
     id:timer
      interval: 10; running: true; repeat: true;
      onTriggered: {
         timeChanged();
         data_arr = info.split(",");
         velocity = parseInt(data_arr[8])
         rpm = parseInt(data_arr[6])
         soc = parseInt(data_arr[7])
         value_left = parseInt(data_arr[1])
         value_right = parseInt(data_arr[0])
         value_lamp1 = parseInt(data_arr[2])
         value_lamp2 = parseInt(data_arr[3])
         value_lamp3 = parseInt(data_arr[4])
         value_brake = parseInt(data_arr[5])
         current= parseInt(data_arr[9])
         if (current > 0xFF) {
            current -= 0xFFF}
          velocity1 = 1.571 + (velocity/180)*(5.814-1.571);
         rpm1 = 2.968 + ((rpm/1000)/20)*(6.460-2.968);
         soc1 = 2.968 + (soc/100)*(6.460-2.968);
         vsupply1 = 2.619 + (vsupply/12)*(0.523-2.619);
         throttle1 = 2.619 + (throttle/100)*(0.523-2.619);
         canvas1.requestPaint();
         canvas2.requestPaint()
```

```
}
Item {
    id: item
    x: 0
    y: 0
    width: 1280
    height: 720
    Canvas {
        id: canvas1
        width: 1280
        height: 720
        anchors.left: parent.left
        anchors.top: parent.top
        antialiasing: true
        onPaint: {
            // Get drawing context
            var ctx = getContext("2d");
            var gradient = ctx.createLinearGradient(0,300,150,0)
            gradient.addColorStop(0, "green")
            gradient.addColorStop(0.4, "yellow")
            gradient.addColorStop(0.8, "red")
            ctx.reset();
            ctx.beginPath();
            ctx.lineWidth = 40;
            ctx.strokeStyle = gradient
            ctx.arc(640, 360, 241, 1.571, velocity1, false)
            ctx.stroke();
                //Left
                if(value_left == 1) {
                    ctx.beginPath();
                    ctx.fillStyle = "green";
                    ctx.fillRect(316.49, 238.80, 55, 40);
                }
```

LAMPIRAN 4 Skematik Mainboard IVC



LAMPIRAN 5 ***Printed Board Circuit*** **Mainboard IVC**



(Halaman ini sengaja dikosongkan)

BIODATA PENULIS

Grangsang Sotyaramadhani dilahirkan di Malang, 19 Maret 1991. Anak kedua dari Agus Susetya dan Eny Pangestuti. Penulis menyelesaikan masa studi sekolah dasar di SDN 1 Sumorame pada tahun 2003, dilanjutkan ke SMPN 1 Candi lulus pada tahun2006 dan SMAN 1 Sidoarjo lulus pada tahun 2009.

Selepas SMA penulis melanjutkan studinya di Institut Teknologi Sepuluh Nopember Jurusan Teknik Mesin pada tahun ajaran 2009/2010. Selama kuliah di ITS penulis aktif mengikuti organisasi Lembaga Bengkel Mahasiswa Mesin. Tidak hanya itu penulis juga telah bergabung dalam tim mobil surya ITS yang telah berkompetisi di Australia pada *World Solar Challenge 2013* serta turut aktif dalam tim Molina (Mobil Listrik Nasional) ITS. Di teknik mesin penulis memilih untuk masuk Laboratorium Otomasi dan mengerjakan tugas akhir dengan topik Perancangan Monitoring System dibawah bimbingan Dr. Muhammad Nur Yuniarto. Pada tahun 2015 penulis menyelesaikan studi S1-nya

(Halaman ini sengaja dikosongkan)